HEWLETT PACKARD

User's Guide

64775

HPC Emulator/Analyzer PC Interface

HEWLETT PACKARD

HP 64775

HPC Emulator PC Interface

Notice

Hewlett-Packard makes no warranty of any kind with regard to this material, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. Hewlett-Packard shall not be liable for errors contained herein or for incidental or consequential damages in connection with the furnishing, performance, or use of this material.

Hewlett-Packard assumes no responsibility for the use or reliability of its software on equipment that is not furnished by Hewlett-Packard.

© Copyright 1991, Hewlett-Packard Company.

This document contains proprietary information, which is protected by copyright. All rights are reserved. No part of this document may be photocopied, reproduced or translated to another language without the prior written consent of Hewlett-Packard Company. The information contained in this document is subject to change without notice.

HP Part No. 64775-97005
First Edition, July 1991
Printed in U.S.A.

HPC is a trademark of National Semiconductor Corporation. IBM and PC, PC AT are registered trademarks of International Business Machines Corporations. MS-DOS is a trademark of Microsoft Corporation. UNIX is a registered trademark of UNIX Systems Laboratories, Inc.

Hewlett-Packard Company
Logic Systems Division
8245 North Union Boulevard
Colorado Springs, CO 80920, U.S.A.

RESTRICTED RIGHTS LEGEND. Use, duplication, or disclosure by the U.S. Government is subject to restrictions set forth in subparagraph (C) (1) (ii) of the Rights in Technical Data and Computer Software Clause at DFARS 252.227-7013. Hewlett-Packard Company, 3000 Hanover Street, Palo Alto, CA 94304

Printing History

New editions are complete revisions of the manual. The date on the title page changes only when a new edition is published.

A software code may be printed before the date; this indicates the version level of the software product at the time the manual was issued. Many product updates and fixes do not require manual changes, and manual corrections may be done without accompanying product changes. Therefore, do not expect a one-to-one correspondence between product updates and manual revisions.

Edition 1 **64775-97005, July 1991**

The HP 64775 HPC Emulator

Description

The HP 64775 emulators and emulation bus analyzer provide real-time, transparent emulation and analysis for National Semiconductor's HPC family of 16-bit CMOS microcontrollers. The emulators support the HPC 16003/083 and 16004/064 to clock rates of 30 MHz. The HPC 16400E communications processor is supported up to a 20 MHz clock rate. The Hewlett-Packard emulation system for the HPC microcontrollers makes the design, test and integration of software and hardware easier than ever before. An easy to use interface, full symbolic support, comprehensive breakpointing and the ability of the emulator to view on-chip bus activity will accelerate the development of your HPC design.

Compatible software development tools from National Semiconductor include an ANSI C Compiler (CCHPC), Assembler/Linker/Librarian (ASMHPC, LNHPC, and LIBHPC) and Source Debugger. The National HPOMFHPC utility generates the file format necessary to work with the HP 64775 emulator.

The Hewlett-Packard PC Interface is an easy to use windowed interface for MS-DOS users. The interface includes windows for Code, Emulation, Trace, Symbols, Error Log, Terminal, and up to four user-defined windows. Several features in the user interface will save you time. Command files can automatically load or store the emulation environment and specific measurements. You won't have to spend your time fumbling through configuration files and measurement set-ups. Instead, you will have time to focus on the problems at hand. Once in the PC Interface, the use of keystroke macros will simplify any repetitive commands you set-up. You can speed the development of your application with the powerful, yet easy to use PC Interface from Hewlett- Packard.

Emulation Features

- Real-time, transparent operation from 2 to 30 MHz for the 16003/083 and 16004/64
- Real-time, transparent operation from 2 to 20 MHz for 16400E
- All of the on-chip I/O ports are recreated. The emulator allows you to "see" bus activity between the CPU, on-chip RAM, on-chip peripherals, and on-chip DMAs.
- Emulation monitor program for display/modify of processor registers and memory, single step, setting software breakpoints, and is modifiable with user-supplied routines.
- Extensive breakpoint capability utilizing the 8 hardware comparators in the emulation bus analyzer for hardware breakpoints and the 32 software breakpoints.
- Support for the standard bank-switch model (AN 497 January, 1988). In expanded memory applications, addresses can be treated as a 4-bit bank code plus a 16-bit offset.
- Full symbolic debug support for fast development and debug with the easy to use PC User Interface.
- Fully coordinated measurement bus for synchronizing or cross triggering up to 32 HP 64700 series emulators or instruments.

Analysis Features

- A 48 channel x 1024 state Emulation Bus Analyzer for real-time analysis of of on-chip HPC address, data and status information.
- Dual-bus architecture for display and modification of trace measurements without halting real-time execution.
- Disassembly of the HPC instruction set in the trace listing.
- Symbols inserted in the trace list let you quickly understand the execution of your code.
- Dequeuing of the microcontroller prefetch to eliminate extraneous cycles in the trace list when the instructions are not executed.
- Tracing of on-chip DMA activity.
- Time tagging of stored states with 40 nsec resolution to measure instruction execution times, bus cycle times, and module duration.
- Eight levels of sequencing (if, then, else).
- Innovative prestore analysis measurement.
- Address and data ranges can be used in trace trigger/store specifications to permit tracing of address and data ranges.

Memory Features

- 128 Kbytes of real-time emulation memory for off-chip memory support to 30 MHz (20 MHz for HPC16400E).
- 32 Kbytes of emulation memory for 20 MHz, 0 wait state on-chip memory support of the 16003/083 and 16004/064.
- Emulation memory support for 16-bit on-chip memory, and 8-bit or 16-bit off-chip memory.
- Memory mapping attributes include Target or Emulation, and RAM, ROM, or GUARDED memory.
- All memory mappable in 256-byte blocks.

In This Book

Chapter 1, "Use the PC Interface," tells you how to set up the emulator and software and how to begin using the PC Interface commands.

Chapter 2, "Use the Emulator," shows you how to use the PC Interface commands to control the emulation processor and make simple emulation measurements.

Chapter 3, "Use the Analyzer," explains how to use the emulation analyzer to record program execution for debugging.

Chapter 4, "Configure the Emulator," explains how to use the PC Interface to allocate emulation resources such as memory and how to enable and disable certain emulator features.

Chapter 5, "Use the Absolute File Reader," explains how to use the absolute file readers. The readers operate on HP64000 format absolute files and produce an absolute file and a symbol database.

Chapter 6, "Solve Problems," explains some of the problems that you might encounter when you use the emulator, and how to solve them.

Chapter 7, "Installation and Service," tells you how to install the emulator in the HP 64700 Series Card Cage. It also gives information on testing and servicing the emulator.

Chapter 8, "Emulator Messages," lists the PC Interface error and status messages that are unique to the HPC emulator and also lists their meanings.

Chapter 9, "HPC Emulator Syntax," shows the syntax of commands and expressions that are unique to the HPC emulator.

Chapter 10, "Specifications and Characteristics," lists the electrical specifications and characteristics for the HPC emulator.

Contents

1 Using the PC Interface 15

To use the emulator 16
To install the emulator and analyzer into the Card Cage 16
To connect the HP 64700 Card Cage to the PC 17
To install the PC Interface software 18
To apply power to the HP 64700 Card Cage 19
To start the PC Interface 19
To select commands 21
To exit the PC Interface 21

Using Windows 22

To create a new user window 22
To delete a user window 23
To make a window the active window 24
To hide a window 25
To view a window 25
To erase the contents of a window 26
To load a file into a window 27
To store the contents of a window to a file 27
To search a window for a string 29
To zoom a window 30
To change window parameters 31
To change window colors 32
To move around in a window 33

Accessing the Terminal Interface 34

To access the Terminal Interface 34
To get help on Terminal Interface commands 35

Accessing the Operating System 37

To enter a single MS-DOS command 37
To enter multiple MS-DOS commands 38

2 Using the Emulator 39

To configure the emulator 40
To build HPC programs 40
To display the emulator status 41

Loading and Storing Programs 43

To load a program 43
To store a program 44

Using Symbols 46

To load a symbol database 47
To display global symbols 48
To display local symbols 48
To transfer global symbols to the emulator 49
To remove global symbols from the emulator 49
To transfer local symbols to the emulator 50
To remove local symbols from the emulator 51

Accessing Processor Memory Resources 52

To display memory as byte values 52
To display memory as word values 53
To display memory as instruction mnemonics 53
To display memory repetitively 54
To modify memory 54
To search memory 56

Using Processor Run Controls 58

To run a program 58
To break to monitor 59
To step the processor 60
To change the step function 61
To reset the processor 61

Viewing and Modifying Registers 63

To display registers 63
To modify registers 65

Using Software Breakpoints 66

To add software breakpoints 66

To set software breakpoints 67
To clear software breakpoints 68
To remove software breakpoints 69
To display software breakpoints 70

Making Coordinated Measurements 71

To start two emulators simultaneously 71

Using the Emulator In-Circuit 72

To prepare for in-circuit emulation 72
To install the target system probe 73
To connect emulator reset to target system reset 74
To install the PLCC to PGA adapter 75
To power-on the emulator and target system 75

3 Using the Analyzer 77

Making Basic Analyzer Measurements 79

To start a trace measurement 79
To halt a trace measurement 79
To display the trace status 80
To display the trace list 80
To define a simple trigger qualifier 80
To define a storage qualifier 81
To set the trigger position 82

Displaying the Trace List 84

To change the trace format 84
To display the trace list 85
To change the trace depth 86

Making Complex Measurements 87

To insert a sequence term 87
To remove a sequence term 88
To assign the trigger term 89
To reset the analyzer 89
To define a qualifier 90
To create an expression 91
To define trace patterns 92

To define a range 94
To set the storage qualifier 94
To define a primary branch term 95
To define a global restart term 96
To define a secondary branch term 96

Using Other Analyzer Features 98

To define a prestore qualifier 98
To count states or time 99

4 Configuring the Emulator 101

Configuring the Target System Interface 105

To select the processor clock source 105
To enable or disable target system resets 105
To configure non-maskable interrupt sharing 106

Configuring and Mapping Memory 109

To set the access mode 109
To enable fast emulation of on-chip memory 111
To set the bus configuration 112
To enable ROMless mode 113
To enable/disable bank switching 114
To define the bank switch address range 115
To assign memory map terms 116
To assign the memory map default 119
To check the memory map 120
To delete memory map terms 120
To reset the memory map 121
To select the emulation monitor 121
To define user monitor routines 123

Defining Break Conditions 125

To define the software breakpoint vector 125
To enable or disable write to ROM break 127
To enable an analyzer hardware break 128
To enable CMB interaction 129
To restrict to real-time runs 129

Setting the Analyzer Configuration 131
To enable tracing of external DMA activity 131
To enable tracing of user NMI monitor processing 132

Using Command Files for Common Configurations 133
To set the configuration for in-circuit emulation 133
To set the configuration for in-circuit emulation and zero-wait state emulation memory 134
To set the configuration for out-of-circuit emulation 134
To set the configuration for out-of-circuit emulation with bank switching 135

5 Using the Absolute File Reader 137

6 Solving Problems 143

If the PC Interface won't start 144
If the emulator won't run 145
If the emulator won't run from reset 146
If the emulator won't run from the monitor 146
If a register or target memory modify doesn't appear to work properly . . 148
If you can't set a software breakpoint or modify the reset vector 148
If you see unexplained states in the trace list 148
If you are having problems running with 8-bit memory 149
If you have problems disassembling a JID or JIDW instruction in memory 149
If you need to step over and not through interrupt service routines 150
If you can't trigger the analyzer 150
If the emulator moves your code or data when you modify the emulator memory map 150
If a nested break into the monitor occurs 151

7 Installation and Service 153

To install the emulator into the HP 64700A Card Cage 157
To install the emulator probe into a PLCC socket 158
To install the emulator probe into a PGA socket 159
To remove the probe cable from the emulation control card 160
To verify the performance of the emulator 161
To insure software compatibility 164
Parts List 166

8 Emulator Messages 169

9 HPC Emulator Syntax 181

Topics Covered 182
ADDRESS 183
demo 185
EQUATES 191
REGISTERS 192

10 Specifications and Characteristics 203

General Emulator Specifications 204

Index

1

Using the PC Interface

How to set up the emulator, start the PC Interface, and enter commands

The PC Interface allows you to use the HPC emulator and analyzer through a simple guided interface. You are prompted to select commands and fill in forms to operate the emulation and analysis functions.

To use the emulator

1 Develop your target system programs as described in chapter 2.

2 Set up the emulator hardware and software as described in this chapter and in chapter 7.

3 Connect the emulator to the demo board or other target system. (See this chapter, chapter 2 and chapter 7.)

4 Apply power to the emulator.

5 Start the PC Interface software (see this chapter).

6 Configure the emulator as needed for the target system and your programs. See chapter 4.

Use the PC Interface commands to load, run and debug your programs. See chapters 1, 2 and 3.

To install the emulator and analyzer into the Card Cage

- Install the emulator and analyzer boards according to the instructions in chapter 7 of this manual.

If you purchased the emulator as a complete system, it is presinstalled in the HP 64700 Series Card Cage at the factory. However, if you purchased only the emulator subsystem, you must install the emulator and analyzer boards in the HP 64700 Series Card Cage before you can use the emulator.

To connect the HP 64700 Card Cage to the PC

1 Set all of the COMM config switches on the rear panel of the Card Cage to 0.

2 Connect a "modem" cable (no wiring reversals) from PORT A on the rear panel of the Card Cage to COM1 or COM2 on the PC. (COM3 or COM4 can be used, but the 64700tab file must be modified as described in the *HP 64700 Series Card Cage Installation/Service Guide.*)

3 Apply power to the HP 64700 Series Card Cage by pressing the front panel power switch to ON. Allow a few seconds for the built-in testing to complete.

4 Install the PC Interface software as described in this chapter. Then start the PC Interface software. If you do not see the initial PC Interface screen after a few seconds, recheck your installation. (The initial PC Interface screen is a set of three stacked windows with a row of commands at the bottom.)

The above steps summarize the procedure for connecting the HP 64700 Card Cage to the PC and testing the installation. More information (including special installation cases) is provided in the *HP 64700 Series Card Cage Installation/Service Guide.*

The PC Interface must be able to communicate with the HP 64700 emulator before you can make emulator measurements. This communication can be over RS-232 or RS-422 serial data links.

To install the PC Interface software

1 Insert disk 1 of the PC Interface software into a floppy drive on your PC.

2 Switch to this drive by typing either A: or B: (depending on the name of the drive that contains the PC Interface floppy). Press **<Enter>**. You must switch to the correct drive before installing the PC Interface software, or the software will not install correctly.

3 Type **install** at the MS-DOS prompt. Press **<Enter>**.

4 Follow the instructions on the screen to complete the installation.

Caution

The installation program can create only one level of subdirectory at a time (this is a limitation of MS-DOS). Therefore, it can create the default subdirectory \HP64700, then install subdirectories and files below it. However, if it cannot create the directory \HP64700, it will display an error message when trying to create subdirectories such as \HP64700\BIN.

5 If you used the default directory to install the PC Interface software, use a text editor (such as EDLIN) to edit the AUTOEXEC.BAT file on your startup disk. You must add the following information to the file:

```
PATH=<existing_path>;C:\HP64700\BIN;
SET HPTABLES=C:\HP64700\TABLES
SET HPBIN=C:\HP64700\BIN
```

If you did not use the default directory to load these applications, you will need to substitute the appropriate path for C:\HP64700.

The file CONFIG.SYS on your startup disk must contain the following statement:

```
FILES=20
```

6 Remove the floppy disk and reboot your system by pressing <Ctrl><Alt><Del> simultaneously.

The PC Interface software must be installed and configured on your PC before you can use the software with the emulator. The above steps summarize the software installation process. If you need more information, see the *HP 64700 Series Card Cage Installation/Service Guide.*

To apply power to the HP 64700 Card Cage

- Apply power to the emulator by pressing the power ON button located on the front panel of the HP 64700 Card Cage.

  You must apply power to the emulator before you start the PC Interface software. Otherwise, the PC Interface can't communicate with the emulator, and won't start.

To start the PC Interface

- At the MS-DOS command prompt, type **pchpc <options> <port_name>** to start the PC Interface without external timing analyzer support.

  or

- At the MS-DOS command prompt, type **pthpc <options> <port_name>** to start the PC Interface with external timing analyzer support.

pchpc and **pthpc** are MS-DOS programs that provide the PC Interface functions. <options> are as follows:

Option	Meaning
/m	Tells the PC Interface that you're using a monochrome monitor.
/w <filename>	Loads a PC Interface configuration file designated by <filename>.
/c <filename>	Loads and executes a PC Interface command file designated by <filename>.
/?	Displays usage information for the PC Interface startup command.

<port_name> is the name of the communications port assigned in the file \hp64700\tables\64700tab.

When you start the PC Interface, it initializes the emulator as necessary (depending on the last exit command) and presents the initial PC Interface windows. If the startup is unsuccessful, you'll be returned to the MS-DOS prompt.

See the *PC Interface Reference* for more information about command and configuration files.

Examples

To start the HPC PC Interface for a system with a monochrome monitor and the emulator attached to the serial port COM1, type:

```
pchpc /m emul_com1
```

To start the HPC PC Interface with timing support, a color monitor and the emulator attached to serial port COM3, and load a configuration file named config1.cfg, type:

```
pthpc /w config1.cfg emul_com3
```

To select commands

- Press the key corresponding to the first letter of the command option you want.

  or

  Use the right and left arrow keys to highlight a command option. Press the **<Enter>** key to select that option.

- Press the **<Esc>** key to retrace the command tree when you select the wrong option.

  When a command or option is highlighted, the bottom line of the display shows the next level of options or a short message describing the current option.

Example Press the keys "p," "g," and "p" to select the command Processor Go Pc.

To exit the PC Interface

- To exit and restore the current configuration the next time you enter the PC Interface, select System Exit Locked.

- To exit and start with the default configuration the next time you enter the PC Interface, select System Exit Unlocked.

- To exit the PC Interface without saving the current configuration, select System Exit No_Save.

  See the *PC Interface Reference* for a complete description of the system exit options and their effect on the emulator configuration.

Using Windows

A window is a view of your input data and measurement results. Each window is a part of the PC Interface display that allows you to observe emulation and analysis operations.

You can use the PC Interface windows to perform multiple operations simultaneously and help organize your measurements. You can execute one function in one window and a different function in another window.

There are two types of windows in the PC Interface: system windows and user-defined windows. System windows are used for emulation and analysis measurement results. User-defined windows allow you to define custom windows for displaying code or data from files on your PC.

This section explains how to perform some common operations using the window commands.

To create a new user window

1 Select **Window Open**.

2 Type a window name of 12 characters or less. Press **<Enter>**.

3 Type a number in the range 1..20 to position the top edge of the window, then press **<Enter>**. (Or, simply press **<Enter>** to accept the default position.)

4 Type a number in the range 1..20 to position the bottom edge of the window, then press **<Enter>**. (Or, simply press **<Enter>** to accept the default position.)

5 Now the cursor is in the "Autoclear" field. Type **y** if you want information that is written to the window to overwrite existing data in the window. Type **n** if you want information written to the window to be appended to the existing data. Press **<Enter>**.

6 Type in a number in the range 20..1030 to specify the buffer size of the window. (Note that each 20 lines in the window buffer consumes about 2 Kbytes of system memory.) Press **<Enter>**.

7 Type a number in the range 0..79 to position the left edge of the window, then press **<Enter>**. (Or, simply press **<Enter>** to accept the default position.)

8 Type a number in the range 0..79 to position the right edge of the window, then press **<Enter>**. (Or, simply press **<Enter>** to accept the default position.)

9 Now the cursor is in the "Display" field. Type **y** if you want the window to be displayed when you finish this procedure. Type **n** if you want the window to be hidden. (You can display the window with the Window Utility View command. See "To view a window.")

10 To save the changes you have made and create the new window, press **<Enter>** when the cursor is in the last field of the window definition form. (You can quickly move to the last field of the form by pressing **<End>**.) Press **<Esc>** to discard your changes and exit the window definition form.

A user-defined window provides a unique window into which you may load source code files or other information that will help you with your system debugging problems.

Since the window definition screen is a form, you may use the arrow keys to move to only the fields you want to change.

To delete a user window

1 Select **W**indow **D**elete.

2 Use **Tab** or **Shift-Tab** to select the name of the user-defined window you wish to delete.

or

Type in the name of the user-defined window you wish to delete.

3 Press **<Enter>** to confirm the name selection.

4 You are asked to confirm the deletion. Type **y** if you want to delete the selected user-defined window. Type **n** to abort the deletion.

5 Press **<Enter>** to delete the window, or press **<Esc>** to abort the command and return to the PC Interface.

When you are finished with a user-defined window, you may wish to delete it. Deleting a user-defined window removes it from the PC Interface display and frees the system memory associated with corresponding window buffer.

To make a window the active window

1 Select **Window Activate**.

2 Use **Tab** or **Shift-Tab** to select the name of the window you want to activate. (The currently active window is the default.)

or

Type in the name of the window you wish to activate.

3 Press **<Enter>** to confirm your selection and activate the window. Press **<Esc>** to abort the window activation and return to the PC Interface screen.

The active window is the one highlighted by a bold border on the PC Interface screen. This is the one that is acted on by PC Interface window commands, such as **Window Zoom**, or by the cursor movement keys.

You can also use the **Ctrl-A** key combination to activate the next available window in the PC Interface display. Keep pressing **Ctrl-A** until the window you want is activated.

To hide a window

1 Select Window Utility Hide.

2 Use **Tab** or **Shift-Tab** to select the name of the window you want to hide. (The currently active window is given as the default.)

or

Type in the name of the window you wish to hide.

3 Press **<Enter>** to confirm your selection and hide the window. Press **<Esc>** to abort the process and return to the PC Interface screen.

You can hide a window to temporarily remove it from the PC Interface screen. This can help clarify the display. The window definition still exists, and data is written to the window as if it were still displayed.

You can't activate a hidden window. You must first make it visible with the Window Utility View command.

To view a window

1 Select Window Utility View.

2 Use **Tab** or **Shift-Tab** to select the name of the window you want to view. Only windows which are currently hidden are listed.

or

Type in the name of the window you wish to view.

3 Press **<Enter>** to confirm your selection and view the window. Press **<Esc>** to abort the process and return to the PC Interface screen.

You can't see the contents of a hidden window, though the PC Interface writes data to hidden windows. To reveal a hidden window and its contents, you use the **Window Utility View** command.

You also need to use this command if a window is hidden and you want to activate it.

To erase the contents of a window

1 Select **Window Erase**.

2 Use **Tab** or **Shift-Tab** to select the name of the window whose contents you want to erase. (The currently active window is the default.)

or

Type in the name of the window.

3 Press **<Enter>** to confirm the name selection.

4 You are asked to confirm the erasure. Type **y** if you want to erase the contents of the selected window. Type **n** to abort the erasure.

5 Press **<Enter>** to erase the window contents, or press **<Esc>** to abort the erase command and return to the PC Interface.

Sometimes you may want to erase the contents of a window to remove incorrect measurement results, or to remove a file that you loaded and no longer need. You use the **Window Erase** command to do this. You can also use the key combination **Ctrl-E** to erase the currently active window, but you are not prompted to confirm the erasure.

To load a file into a window

1 Select Window Load.

2 Use **Tab** or **Shift-Tab** to select the name of the window into which you want to load a file. (The currently active window is the default.)

or

Type in the name of the window.

3 Press **<Enter>** to confirm the name selection.

4 Type in the name of a file to load into the window. You should load only ASCII text files. The results of loading non-text file types are unpredictable. You can include a drive specifier, directory path, and file extension if desired.

5 Press **<Enter>** to load the file contents into the selected window. Press **<Esc>** to abort the load command and return to the main PC Interface screen.

You can load a file into a window to view the file's contents. This is handy for viewing source code files while debugging. Or, you might load files that describe system specifications while you are debugging that section of the system.

To store the contents of a window to a file

1 Select Window Store.

2 Use **Tab** or **Shift-Tab** to select the name of the window whose contents you want to store. (The currently active window is the default.)

or

Type in the name of the window.

3 Press **<Enter>** to confirm the name selection.

4 Type in the number of the first line in the range of lines you want to store.

or

Position the cursor in the window on the first line in the range of lines you want to store. Use the up and down arrow keys to do this. The line number will be reflected in the "From line" field of the form.

Press **<Enter>** to move to the next field.

5 Type in the number of the last line in the range of lines you want to store.

or

Position the cursor in the window on the last line in the range of lines you want to store. Use the up and down arrow keys to do this. The line number will be reflected in the "Thru line" field of the form.

Press **<Enter>** to move to the next field.

6 Type in the name of a file into which the window contents should be stored. You can include a drive specifier, directory path, and file extension if desired.

7 Press **<Enter>** to store the window contents into the selected file. Press **<Esc>** to abort and return to the main PC Interface screen.

You may want to store the results of a measurement for later use. The **W**indow **S**tore command allows you to do this. By specifying a range of lines to be stored, you can limit the information stored to that which interests you.

You can't store an empty window to a file. If you specify the name of an existing file, the window contents are appended to the contents of the file.

Example

To print lines 20 through 30 of the Symbols window on a system printer attached to your PC, enter:

Window Store Symbols **20 <Enter> 30 <Enter> prn <Enter>**

To search a window for a string

1 Select **Window Utility Search**.

2 Use **Tab** or **Shift-Tab** to select the name of the window whose contents you want to search. (The currently active window is the default.)

or

Type in the name of the window.

3 Press **<Enter>** to confirm the name selection.

4 Type in the number of the first line in the range of lines you want to search.

or

Position the cursor in the window on the first line in the range of lines you want to search. Use the up and down arrow keys to do this. The line number will be reflected in the "From line" field of the form.

Press **<Enter>** to move to the next field.

5 Type in the number of the last line in the range of lines you want to search.

or

Position the cursor in the window on the last line in the range of lines you want to search. Use the up and down arrow keys to do this. The line number will be reflected in the "Thru line" field of the form.

Press **<Enter>** to move to the next field.

6 Type in the string that you want to find in the selected window.

7 Press **<Enter>** to search the window contents for the selected string. Press **<Esc>** to abort the command and return to the main PC Interface screen.

If the window buffer is large and the buffer is nearly full, you may find it tedious to page through the window display while searching for a particular measurement value or source line. Instead, you can use the **W**indow **U**tility **S**earch command to find the data.

When the PC Interface finds the string in the window, it places the cursor immediately after the string with the string at the top of the window.

To zoom a window

1 Select **W**indow **Z**oom.

2 Use **Tab** or **Shift-Tab** to select the name of the window that you want to zoom. (The currently active window is the default.)

or

Type in the name of the window.

3 Press **<Enter>** to confirm the selection and zoom the window to full screen size. Press **<Esc>** to abort the window zoom.

Often you'll want to see more information than can be displayed in the default window size, but you don't want to change the window parameters permanently. You can temporarily expand any window to the full PC Interface screen size. Other windows are obscured under the zoomed window. Only one window may be zoomed at a time.

If you perform the zoom operation on a window that already occupies the full PC Interface screen, it will be returned to the size defined by the window parameters.

To zoom the currently active window without using the **W**indow **Z**oom command, press **Ctrl-Z**. To return the active window to its original size, press **Ctrl-Z** again.

To change window parameters

1 Select Window Utility Parameters.

2 Use **Tab** or **Shift-Tab** to select the name of the window that you want to modify. (The currently active window is the default.)

or

Type in the name of the window.

3 Press **<Enter>** to confirm the name selection.

4 Type a number in the range 0..20 to position the top edge of the window, then press **<Enter>**. (Or, simply press **<Enter>** to accept the default position.)

5 Type a number in the range 0..20 to position the bottom edge of the window, then press **<Enter>**. (Or, simply press **<Enter>** to accept the default position.)

6 Now the cursor is in the "Autoclear" field. Type **y** if you want information that is written to the window to overwrite existing data in the window. Type **n** if you want information written to the window to be appended to the existing data. Press **<Enter>**.

7 Type in a number in the range 20..1030 to specify the buffer size of the window. (Note that each 20 lines in the window buffer consumes about 2 Kbytes of system memory.) Press **<Enter>**.

8 Type a number in the range 0..79 to position the left edge of the window, then press **<Enter>**. (Or, simply press **<Enter>** to accept the default position.)

9 Type a number in the range 0..79 to position the right edge of the window, then press **<Enter>**. (Or, simply press **<Enter>** to accept the default position.)

10 Now the cursor is in the "Scroll" field. Type **y** if you want the PC Interface to write data to the window one line at a time as it becomes available. (Usually more useful for large trace listings or displaying large source files.) Type **n** if you want the PC Interface to wait until it has all requested data available before it writes the data to the window. (The second method is faster overall, but you may notice an initial delay while the PC interface gathers the information.)

11 To save the changes, press **<Enter>** when the cursor is in the last field of the window parameters form. (You can quickly move to the last field of the form by pressing **<End>**.) Press **<Esc>** to discard your changes and exit the window parameters form.

You may find that the default window definitions don't meet your needs. You can use the Window Utility Parameters form to change the window definition to your tastes. Also, this form is the only way that you can change the way in which data is written to the window (the "Scroll" parameter).

Example

To reposition and resize the Symbols window, enter the following command:

```
Window Utility Parameters Symbols <Enter> 0 <Enter> 12
<Enter> y <Enter> 48 <Enter> 30 <Enter> 79 <Enter> n
<Enter>
```

To change window colors

1 Select Window Utility Color.

2 Use **Tab** and *Shift-Tab* to select **color** if you have a display card and monitor that are capable of displaying color. Select **mono** if either your display card or monitor is restricted to monochrome.

3 Use **Tab** or **Shift-Tab** to select the foreground color from the following choices:

white, black, blue, green, cyan, red, magenta

Press **<Enter>**.

4 Use **Tab** or **Shift-Tab** to select the background color from the above choices.

5 Press **<Enter>** to save your changes, or press **<Esc>** to abort the command and leave the colors unchanged.

Changing colors is useful only if you have a color monitor. Although the mono monitor choice will allow you to change color selections, the display will always have black background color and white foreground color.

The PC Interface will not allow you to set the background and foreground colors the same.

To move around in a window

- To move the cursor up in the window, use the up arrow key.
- To move the cursor down in the window, use the down arrow key.
- To move to the next page in the window buffer, use the **Pg Dn** key.
- To move to the previous page in the window buffer, use the **Pg Up** key.
- To move the cursor to the right in the window, use **Ctrl-right arrow**.
- To move the cursor to the left in the window, use **Ctrl-left arrow**.
- To move to the beginning of the window buffer, press the **Home** key.
- To move to the end of the window buffer, press the **End** key.

Accessing the Terminal Interface

You can control the emulator with a set of commands that reside in the emulator's firmware. This command set is called the Terminal Interface. The Terminal Interface is described in the *MC68020/MC68EC030 Emulator Terminal Interface User's Guide* and in the *Terminal Interface Reference*.

This section explains how to access the Terminal Interface from within the PC Interface.

To access the Terminal Interface

- To enter the Terminal Interface, select **System Terminal**.
- To leave the Terminal Interface, press **Ctrl-**.

The Terminal Interface allows access to the low-level commands of the emulator. You need to use caution when using the Terminal Interface commands, because the PC Interface operation may be disturbed by changes to the emulator configuration using the Terminal Interface. Do not use commands that change the emulator's data communications parameters.

Commands	Reason to Avoid
stty, po, xp	Do not use. Will change the channel operation and hang emulator.
echo, mac	Usage may confuse the channel protocol.
wait	Do not use, will block access to emulator.
init, pv	Will reset emulator configuration.
t	Do not use. Will confuse trace status polling and unload.
tcf -e	Do not use unless you are planning to run pv. The PC Interface requires the complex mode (tcf -c).

See the *HPC Emulator Terminal Interface User's Guide* and the *Terminal Interface Reference* for more information regarding the Terminal Interface.

Examples

To access the Terminal Interface and display memory locations 0 through 20 in word format, enter:

```
System Terminal
m -dw 0..20
Ctrl-\
```

To access the Terminal Interface and check the emulator status, then the trace configuration, enter:

```
System Terminal
es
tcf
Ctrl-\
```

To get help on Terminal Interface commands

1 Select System Terminal.

2 Enter: **help <cmd_name>**

where **<cmd_name>** is the Terminal Interface command on which you want to receive help.

See the *HPC Emulator Terminal Interface User's Guide* and the *Terminal Interface Reference* for more information regarding the Terminal Interface.

Examples

To get help on the Terminal Interface **cf** command, enter:

```
System Terminal
help cf
Ctrl-\
```

To get help on all Terminal Interface command groups, enter:

```
System Terminal
help *
Ctrl-\
```

Accessing the Operating System

The PC Interface includes commands that allow you to perform simple MS-DOS operating system functions without leaving the PC Interface. You can enter a single command, then return immediately to the PC Interface, or enter a series of commands at the MS-DOS prompt.

When you use the operating system access capability, you should not run any programs that modify the communications ports (COM1-COM4) (such as terminal emulation programs). Also, you should not run any programs that consume system memory and do not release it. Many network management programs and TSR (terminate-and-stay-resident) programs fit this definition.

Also, there is a limited amount of memory available when you use the system commands from the PC Interface. You cannot run programs that require large amounts of system memory, such as compilers or complex text editors.

This section explains how to use the system access commands.

To enter a single MS-DOS command

1 Select System MS-DOS Command.

2 Type in the MS-DOS command you want to execute (include any command parameters such as filenames or command options).

3 Press **<Enter>** to execute the command. Press **<Esc>** to discard the command and return to the PC Interface.

4 When the command has finished executing, press any key to return to the PC Interface.

Examples

To display all files in the current directory, enter:

```
System MS-DOS Command dir <Enter>
```

To format a floppy in the A: drive, enter:

```
System MS-DOS Command format a: <Enter>
```

To enter multiple MS-DOS commands

1 Select System MS-DOS Fork.

2 An MS-DOS prompt will appear. Type in the desired series of MS-DOS commands.

3 When you are ready to return to the PC Interface, type **exit** at the MS-DOS prompt, then press **<Enter>**.

2

Using the Emulator

How to control program execution and examine system resources

The emulator has many commands and features that allow you to control execution of your program. It also has facilities for entering and recalling commands.

To configure the emulator

- Set up the emulator for use by configuring it as described in chapter 4.

The emulator has several configuration items that adapt it to specific target system designs and program requirements. You should check the configuration and modify it for your needs before using the emulator. This will ensure correct operation of all emulator functions.

To build HPC programs

1 Create source files in HPC assembly language or HPC C language using a text editor.

2 If you created HPC C language files, translate them to HPC assembly language using the National Semiconductor CCHPC compiler.

3 Translate the assembly source files to relocatable object files using the National Semiconductor ASMHPC assembler.

4 Link all relocatable object files with the National Semiconductor LNHPC linker to produce both an absolute object file in the COFF (.cof) format and an absolute object file in the Intel Hex (.hex) format.

Absolute code can be downloaded from any host computer in the Intel hex format without using the HP 64000 **transfer** utility.

5 Convert the COFF (.cof) absolute file to HP 64000 format (one .L file, one.X file, and one .A file for each assembly source file) using the HPOMFHPC utility (supplied by National).

6 Convert the HP64000 format object fileset to a binary absolute file (.hpa) and an ASCII symbol file (.hps) using the RHP64000 file format reader. See the chapter "Use the Absolute File Reader."

If you're planning to load programs into emulation or target system memory, you need to have your files in a format acceptable to the HPC emulator. With the PC Interface, this usually means that you'll want your files in HP64000 absolute format. The National Semiconductor language tools for the IBM PC and compatible computers can produce files in this format. Often Intel hex format will be more useful, especially if you need to download the file to the emulator from a variety of host systems. However, the Intel hex format does not support symbol handling.

Note that in version 2.1 of the National Semiconductor language tools, the source files must be in the same directory as the .L file. Later versions of the language tools should not have this limitation.

Example

The source files for the emulator demo program are installed in the directory \hp64700\demo\64775. Copy these files to your working directory. You can build the demo program as follows:

```
asmhpc /listfile /sym /pw=80 /noheadings /crossref demo
asmhpc /listfile /sym /pw=80 /noheadings /crossref int_serv
lnhpc /crossref /map demo,int_serv /out=demo /format=hex
hpomfhpc demo
rhp64000 demo.x
```

The above set of commands is in the file **demo.bat**, which is placed in the directory **\hp64700\demo\64775** by default.

To display the emulator status

- To see the emulator status, look at the left hand side of the STATUS line near the bottom of the PC Interface display.

The emulation STATUS line can tell you most information about the emulator's status: whether the emulator is reset, running a user program, or running in monitor.

Loading and Storing Programs

The PC Interface provides commands that allow you to move files into emulation or target memory from the PC through the serial ports of the HP 64700 Card Cage. Or, you can save a range of memory in an absolute file for later reuse. (You might do this if you patch a section of code and need to do further testing.)

Many different absolute file formats are supported. The primary one discussed in this section is the HP64000 absolute format.

To load a program

1 Check to make sure that you have mapped memory as needed for your system design. (See the section on mapping memory in chapter 4.)

2 Select **Memory Load**.

3 Use **Tab** and **Shift-Tab** to select the format of your absolute file. Press **<Enter>**.

4 Use **Tab** and **Shift-Tab** to select whether to load emulation memory, target system memory, or both. Press **<Enter>**.

5 If you're using the HP64000 file format, **Tab** to select **yes** if you want to force the file format reader to read the file whether or not it needs to be updated. Press **<Enter>**.

6 Enter the name of your absolute file in the last field. For the HP64000 format, enter the linker symbol filename.

7 Press **<Enter>** to load the absolute file. Press **<Esc>** to discard your entries in the form and return to the PC Interface command line.

The emulator accepts absolute files in the following formats:

- HP64000 absolute.
- Raw HP64000 absolute.
- Intel hexadecimal.
- Tektronix hexadecimal.
- Motorola S-records.

The "Force absolute file read?" option is only available for the HP64000 format. It forces the file format reader to regenerate the emulator absolute file (.hpa) and symbol database (.hps) before loading the code. Normally, these files are only regenerated whenever the file you specify (the output of your language tools) is newer than the emulator absolute file and symbol database. For more information, refer to the File Format Reader chapter.

Example

To load the demo program, select Memory Load. Leave the file format as HP64000, and move to the "File name" field using the arrow keys. Type **demo.l**, and press **<Enter>** to load the demo program. (You must be in the same directory as the demo program, or provide the full pathname to the file.)

To store a program

1 Select Memory Store.

2 **Tab** to select a file format for the stored program data, then press **<Enter>**.

3 Type in a memory range: **<lower>..<upper>** and press **<Enter>**.

4 Type in the name of an absolute file. Press **<Enter>**.

You can save a range of memory locations in an absolute file, which you can reload into the emulator when needed. Suppose you make some changes to a program by modifying memory, and want to continue testing later. You can save the program in an absolute file to ensure that your changes are available for future use.

You can save memory to files in the following formats:

- Raw HP64000 absolute.
- Intel hexadecimal.
- Tektronix hexadecimal.
- Motorola S-records.

It's a good idea to make sure that the filename you specify isn't already in use (unless you want to overwrite the file with that name).

Using Symbols

Symbol handling adds power to your interaction with the HPC emulator. You can use symbols in expressions involving addresses, which frees you from memorizing the addresses associated with the symbols.

HP64000 symbol files are created when you convert absolute files from COFF format to HP64000 format with the National Semiconductor HPOMFHPC utility. (See "To build HPC programs" for more information about this utility and specific limitations in version 2.1 of the National Semiconductor language tools.)

When you convert a .COF file, an assembler symbol file (with the same base name as the source file and a ".A" extension) is created for each source file that can be accessed by HPOMFHPC. The assembler symbol files contain local symbol information. Also, a linker symbol file (with the same base name as the absolute file and a ".L" extension) is created. The linker symbol file contains global symbol information and information about the relocatable assembly modules that combine to form the absolute file.

When you load a file using the HP64000 file format, the absolute file reader collects global symbol information from the linker symbol file and local symbol information from the assembler symbol files. It uses this information to *create* a single symbol database with the extension .hps. The symbol database is automatically created and *loaded* into the PC Interface when you load the absolute file.

The symbol database is accessible only to the PC Interface, and allows you to use symbols in address expressions. For example, you can specify a symbol in a run address or in a memory display command.

Additional symbol capabilities are available when you *transfer* symbol data to the emulator from the PC Interface. After you transfer symbol data to the emulator, you can display memory locations and analyzer trace lists with symbols.

Symbol Scoping

A symbol can be local in one module and global in another, which may cause confusion. For example, suppose symbol "XYZ" is a global in module A and a local in module B and that these modules link to form the absolute file. After

you load the absolute file (and the corresponding symbol database), entering "XYZ" in an expression refers to the symbol from module A. Then, if you display local symbols from module B, entering "XYZ" in an expression refers to the symbol from module B, **not the global symbol.** Now, if you again want to enter "XYZ" to refer to the global symbol from module A, you must display the local symbols from module A (since the global symbol is also local to that module). Loading local symbols from a third module, if it was linked with modules A and B and did not contain an "XYZ" local symbol, would also cause "XYZ" to refer to the global symbol from module A. You can always refer to the global symbol with the syntax :XYZ.

For more information on symbols, refer to the *PC Interface Reference*.

To load a symbol database

- Load a file in HP64000 format using the **Memory Load** command.

  or

- Select **System Symbols Global Load**, type the name of the linker symbol file, and press **<Enter>**.

When you load an HP64000 format absolute file into the emulator, the corresponding symbol database is also loaded. Whenever you load a symbol database, if the database is older than the linker symbol file, the PC Interface runs the absolute file reader to create a new symbol database.

The symbol database also can be loaded with **System Symbols Global Load**. Use this command when your application consists of multiple absolute files. You can load the various symbol databases corresponding to each absolute file. When you load a symbol database, information from a previous symbol database is lost. That is, only one symbol database can be present at a time.

After a symbol database is loaded, both global and local symbols can be used when entering expressions. You enter global symbols as they appear in the source file or in the global symbols display.

To display global symbols

1 Select **System Symbols Global Display**.

2 Press **<Ctrl>Z** to zoom the window if desired.

The symbols window automatically becomes the active window because of this command.

The global symbols display has two parts. The first part lists all the modules that were linked to produce this object file. You use these module names when you want to refer to a local symbol. They are case-sensitive. The second part of the display lists all global symbols in this module. These names can be used in measurement specifications, and are case-sensitive. For example, if you wish to make a measurement using the symbol **CMD_INPUT**, you must specify **CMD_INPUT**. The strings **cmd_input** and **Cmd_Input** are not valid symbol names here.

To display local symbols

1 Select **System Symbols Local Display**.

2 Type the name of the module you want to display (from the first part of the global symbols list) and press **<Enter>**.

3 Press **<Ctrl>Z** to zoom the window if desired.

After you display local symbols with the "**System Symbols Local Display**" command, you can enter local symbols as they appear in the source file or local symbol display. When you display local symbols for a given module, that module becomes the default local symbol module.

If you have not displayed local symbols, you can still enter a local symbol by including the name of the module:

```
module_name:symbol
```

Remember that the only valid module names are those listed in the first part of the global symbols display, and are case-sensitive for compatibility with other systems (such as HP-UX).

When you include the name of an source file with a local symbol, that module becomes the default local symbol module, as with the "System Symbols Local Display" command.

Local symbols must be from modules that form the absolute whose symbol database is currently loaded. Otherwise, no symbols will be found (even if the named assembler symbol file exists and contains information).

To transfer global symbols to the emulator

- Select System Symbols Global Transfer.

You can use the emulator's symbol-handling capability to improve measurement displays. You do this by transferring the symbol database information to the emulator.

The symbols and corresponding address information are stored in an emulator system table. When you display memory in mnemonic form, step the processor, or display trace results, the emulator retrieves the symbol information from the table and displays it. This makes the measurement results easier to read.

To remove global symbols from the emulator

- Select System Symbols Global Remove.

If you work with large symbol databases, you may fill the emulator's system memory. This may cause problems if you try to define new equates or macros using the Terminal Interface, or if you want to load more local symbol

information. You can free memory by removing groups of symbols from the emulator.

After you execute this command, the symbol database still exists for the PC Interface. Thus, you can still use the global symbols in expressions. However, the global symbols will not appear in memory and trace list displays.

To transfer local symbols to the emulator

- Select System Symbols Local Transfer All to transfer local symbol information for all modules in the symbol database.

or

- Select System Symbols Local Transfer Group and type in one or more module names, separated by commas.

As with global symbols, you can transfer local symbol information to the emulator. You can transfer information for all modules in the database at once. Or, to save space in the emulator's system memory, you could transfer only the symbols related to the module(s) used in your measurements.

Examples

To transfer local symbol information for all modules in the demo program, select:

```
System Symbols Local Transfer All
```

To transfer local symbol information from the assembly module int_serv in the demo program, select:

```
System Symbols Local Transfer Group  int_serv.asm
<Enter>
```

To remove local symbols from the emulator

- Select System Symbols Local Remove All to remove local symbol information for all modules in the symbol database.

  or

- Select System Symbols Local Remove Group and type in one or more module names, separated by commas.

When you're finished making measurements using one group of local symbols, you may want to remove them from the emulator to make room for new symbol groups.

After you execute this command, the symbol database stills exists for the PC Interface. Thus, you can still use the local symbols in expressions. However, the local symbols for the modules that you removed will not appear in memory and trace list displays.

Examples

To remove local symbol information for the assembly module int_serv in the demo program, select:

```
System Symbols Local Remove Group  int_serv.asm <Enter>
```

To remove local symbol information for all modules in the demo program, select:

```
System Symbols Local Remove All
```

Accessing Processor Memory Resources

While you are debugging your system, you may want to examine memory resources. For example, you may need to verify that the correct data is loaded, or check to see if a sequence of values was written correctly. Also, you may need to modify one or more memory locations to test different data sets for a program. The emulator has flexible memory commands that allow you to view and modify memory as needed.

To display memory as byte values

1 Select Memory Display Byte.

2 Type in one or more individual addresses or address ranges. If you specify multiple addresses, separate them with semicolons.

3 Press **<Enter>**.

Examples

To display program memory for the main part of the demo program as byte values, select:

```
Memory Display Byte INIT..demo.asm:END_MAIN <Enter>
```

To display the demo program's message output buffer and command input buffer in byte format, select:

```
Memory Display Byte MSG_DEST..MSG_DEST+20;CMD_INPUT
<Enter>
```

MSG_DEST is in the on-chip memory area. You must be running the user program or running in the monitor to display memory locations in this area.

To display memory as word values

1 Select Memory Display Word.

2 Type in one or more individual addresses or address ranges. If you specify multiple addresses, separate them with semicolons.

3 Press **<Enter>**.

Examples

To display program memory for the main part of the demo program as word values, select:

```
Memory Display Word INIT-1..demo.asm:END_MAIN <Enter>
```

(INIT-1 is used because words must be on an even address boundary.)

To display memory as instruction mnemonics

1 Select Memory Display Mnemonic.

2 Type in one or more individual addresses or address ranges. If you specify multiple addresses, separate them with semicolons.

3 Press **<Enter>**.

Examples

To display program memory for the main part of the demo program, select:

```
Memory Display Mnemonic INIT..demo.asm:END_MAIN <Enter>
```

To display program memory for the interrupt service routine, select:

```
Memory Display Mnemonic
HANDLE_CMD..int_serv.asm:RESET_PEND+0d <Enter>
```

To display memory repetitively

1 Select Memory Display Repetitively.

2 When you're finished using the changing memory display, press **<Esc>** to return to the PC Interface command line.

Repetitive memory display can be useful if you want to watch the activity in a certain memory area while a target system program is executing. However, it uses the communication channel from the PC to the emulator continuously, so you can't enter other PC Interface commands while the repetitive display is in process.

The address range and display format are those specified in the last Memory Display command.

To modify memory

1 Select Memory Modify Byte to modify memory locations in byte mode.

or

Select Memory Modify Word to modify memory locations according to words (two bytes).

2 To modify a single memory location to a single value, type:
<address>=<value>

or

To modify a range of memory locations to a single value, type:
<lower>..<upper>=<value>

or

To modify a range of memory locations with a list of values, type:
<lower>..<upper>=<value1>,<value2>,...

3 Press **<Enter>** to start the modify.

The **<address>** parameter is an expression representing a single address location. The **<lower>** and **<upper>** values are address expressions representing the lower and upper boundaries of the memory area to be searched. **<value>** represents the data value to which the contents of memory are to be modified.

The emulator uses the monitor to modify memory locations in the on-chip RAM area. Therefore, you cannot modify memory when the emulator is reset. Either run a user program or break to the monitor.

Examples

To modify the byte at e01f hex to 43, select:

```
Memory Modify Byte 0e01f=43 <Enter>
```

To modify the command input buffer of the demo program to 43 hex, enter:

```
Memory Modify Byte CMD_INPUT=43 <Enter>
```

To modify the range of locations from 1c0 through 1d2 to zero, enter:

```
Memory Modify Byte 1c0..1d2=0 <Enter>
```

To modify the range of locations from 1c0 through 1d2 to "ABC", enter:

```
Memory Modify Byte 1c0..1d2=41,42,43 <Enter>
```

You can combine multiple address ranges and value lists. For example:

```
Memory Modify Byte CMD_INPUT=43;1c0..1d2=41,42,43
<Enter>
```

Remember that the memory modification is affected by the display mode. Suppose that locations 1c0 and 1c1 each contain 01. If you enter the command:

```
Memory Modify Byte 1c1=03 <Enter>
```

Then location 1c0 contains 01 and location 1c1 contains 03. But, if you entered:

```
Memory Modify Word 1c0=3 <Enter>
```

Then location 1c0 will contain 03, and location 1c1 will contain 00. Notice that you refer to a word by an even address, which is the address of its least significant byte in the HPC memory architecture.

To search memory

1 Select Memory Find.

2 Specify an address range in the form **<lower>..<upper>**, then press **<Enter>**.

3 To search a memory range for a particular expression, type: **<expr>**

or

To search a memory range for a character string, type: "**<string>**"

4 Press **<Enter>** to start the search.

Searching memory for values or character strings can help you determine whether a program is functioning correctly. For example, in the emulator demo program, you can enter a command, then search memory for the output message to see if the program responded correctly.

Sometimes you expect a data value to be written to a particular memory location during a program run. But, a design error may cause the program to accidentally write the value to the wrong location. You can search memory for expression to see if the value was written to another location.

The **<lower>** and **<upper>** values are address expressions representing the lower and upper boundaries of the memory area to be searched.

If you're searching for character strings, <string> is an ASCII string delimited by ` (accent grave) or " (double quote). Remember that if one of the characters (accent grave or double quote) is part of the string, you should use the other character as a delimiter.

The emulator uses the monitor to read memory in the on-chip RAM area. Therefore, you can't do searches in this area when the emulator is reset. Use the Processor Break command to begin running in the monitor.

Examples

Suppose that memory location 1c0 contains 00 and 1c1 contains 03 hex. Then the word spanning both locations contains 0300 hex.

To search for the value 3 hex by bytes, enter:

```
Memory Find 1c0..1c1 <Enter> 3 <Enter>
```

The match is found at address 1c1.

To search the message area of the emulator demo program for the string "ommand," enter:

```
Memory Find MSG_A..END_MSGS <Enter> "ommand" <Enter>
```

Three matches are found.

Using Processor Run Controls

When you don't use an emulator, run control can be difficult. Usually, you're limited to starting the processor from reset, then entering data values that vector program execution to the routines you want to test. Reaching those routines may be difficult or impossible if the data values are boundary conditions or the program logic is faulty.

By using the emulator, you can run the processor from the current program counter or any desired address. If you want to examine the state of your system after each program instruction, you can use the Processor Step command to step through the program. You can break to the monitor program to examine on-chip resources such as RAM and registers. And, you can reset the processor from the emulator.

To run a program

- Select Processor Go Pc to run the processor from the current program counter (PC) value.
- Select Processor Go Address and type in an address to run the processor from a specific address value.
- Select Processor Go Reset to run the processor from target system reset.

When you're ready to start a program run, you use the Processor Go command.

The address is a 16-bit address expression. If you're operating the processor in bank-switched mode, you specify the bank code by modifying the bc register. See <ADDRESS> in the "Syntax" chapter for more information.

If you select run from reset, the emulator status shows that it's waiting for a target system reset. The emulator has entered a special background monitor loop. When the target system reset occurs, the emulator will begin running from the address contained in the reset vector. This command allows you to

test response of your target system to a reset signal without adding emulator overhead.

If you have problems with run from reset, see chapter 6, "Solving Problems."

Examples

To run from the demo program's starting locations, select:

```
Processor Go Address INIT <Enter>
```

To run the command interpreter code in the demo program's interrupt service routine, select:

```
Processor Go Address int_serv.asm:INTERPRET_CMD <Enter>
```

(The above example works only when the processor entered the interrupt routine via an interrupt; otherwise, the stack will not contain the correct value to return to the main program.)

If you have a program that uses bank switching, to run from an address 305f in bank 3, you select:

```
Register Modify Single bc <Enter> 3 <Enter>
Processor Go Address 305f <Enter>
```

To break to monitor

- Select Processor Break.

The emulation monitor is a program that provides various emulation functions, including register access and target system memory manipulation. You must break execution to the monitor before executing certain emulation commands, such as those accessing registers, target system or on-chip memory. You also can use the break command to pause user program execution.

The emulator status changes to show that the processor is running in the monitor.

You can use either a foreground or background monitor. See the "Configure the Emulator" chapter for more information.

To step the processor

- Select Processor Step Pc. Type in the number of instructions to step, and press **<Enter>**. This steps from the current program counter value.

  or

- Select Processor Step Address. Type in the number of instructions to step, and press **<Enter>**. Type in the starting address for the step, and press **<Enter>**. This steps from a specific address.

The Processor Step command lets you single-step the processor through program code. You can display registers after each step to help you locate the source of problems or verify correct operation. Or, you might want to modify a register, then step the processor to check the result.

You can specify a step count (<count>) to step the processor more than one instruction. The default base is decimal. The default base for <address> is hex.

If your application is primarily interrupt driven, you may encounter some problems when you use the background monitor. It may be best to configure the emulator for the foreground monitor for these applications. See chapter 6, "Solving Problems."

Examples

To step the processor one instruction, select:

```
Processor Step Pc 1 <Enter>
```

To step the processor three instructions from the current program counter, select:

```
Processor Step Pc 3 <Enter>
```

To step the processor five instructions from the address e045, select:

```
Processor Step Address 5 <Enter> 0e045 <Enter>
```

To change the step function

1 Select Processor Step Events.

2 **Tab** to select **y** if you want to display registers after each step, or **n** if you don't want register display. Press **<Enter>**.

3 **Tab** to select **y** if you want the instruction mnemonic displayed at each step, or **n** if you don't want mnemonic display. Press **<Enter>**.

4 Type the name of a command file to execute when stepping is complete. (This is optional).

5 Press **<Enter>** to accept the changes. Press **<Esc>** to discard the changes.

You can alter the way the step command works to give you only the information you need. This helps avoid display clutter for long step sequences.

Also, you can enhance the step command by specifying an optional command file to execute when stepping. See the *PC Interface Reference* for more information on command files, including command syntax.

To reset the processor

- Select Processor Reset Hold to hold the processor reset from the emulator.

- Select Processor Reset Monitor to reset the processor and start executing in the monitor.

- To reset the emulator from the target system, assert the RESET signal in your target system.

When you apply power to the emulator, the initialization process leaves the emulator in the reset state. Changing some configuration items also resets the processor. (See the "Configure the Emulator" chapter for more information.)

Sometimes you may want to reset the emulation processor prior to a program run. The Processor Reset command allows you to do this. Or, you can reset the emulation processor from the target system.

When you use the emulator in-circuit, you may want to connect the emulator reset on the target system probe to the target system reset. This ensures that the target system is properly reset when the emulator leaves reset and begins running a user program.

A target system reset does not reset the entire emulator. It resets only the emulation processor.

Viewing and Modifying Registers

The emulator allows you to display registers to determine the results of program execution. You can display register either singly or in groups of related registers.

Sometimes, you may want to modify a register, then run a segment of program code to test the results.

To display registers

- Select **Register Display Single**, then use the Tab key to select an individual register to display. Or, you can type in the register name. Press **<Enter>** to accept your selection.

  or

- Select **Register Display Class**, then use the Tab key to select a group of registers to display. Or, you can type in the register class name. Press **<Enter>** to accept your selection.

  or

- Select **Register Display Basic** to display all registers from the processor's basic register set.

The available registers and register classes are as follows:

Register Class	Register Names
basic	pc, a, b, k, x, sp, psw, fp, bc, locz
tmr0	t0con, tmmode, divby, r3, t3, r2, t2, i2cr, i3cr, i4cr
tmr4	eicr, eicon, portp, pwmode, r7, t7, r6, t6, r5, t5, r4, t4
uart	tbuf, rbuf, enur, enui, enu, divby
upi	upic, ibuf, obuf, portb, dirb, bfun

Register Class	Register Names
ints	irpd, enir, ircd, porti, psw, eicon, halten
adc	adr7, adr6, adr5, adr4, adr3, adr2, adr1, adr0, adcr3, adcr2, adcr1, portd
ht1	dmt1bc2, dmt1fa2, dmt1bc1, dmt1fa1, dmt1tc, portd, msgctl, msgpnd, serdec, msgcfg, hd1ctl, hd1est
ht2	dmt2bc2, dmt2fa2, dmt2bc1, dmt2fa1, dmt2tc, portd, msgctl, msgpnd, serdec, msgcfg, hd2ctl, hd2est
hr1	dmr1fl, dmr1da2, dmr1ca2, dmr1da1, dmr1ca1, dmr1rc, portd, msgctl, msgpnd, serdec, msgcfg, hd1crc2, hd1crc1, hd1est, hd1rst, hd1ctl, hd1ra2, hd1ra1
hr2	dmr2fl, dmr2da2, dmr2ca2, dmr2da1, dmr2ca1, dmr2rc, portd, msgctl, msgpnd, serdec, msgcfg, hd2crc2, hd2crc1, hd2est, hd2rst, hd2ctl, hd2ra2, hd2ra1
400e	portri, portro, dirr, udiv, udivr
uwire	sio, bfun, dirb, ircd, irpd, divby
ports	porta, portb, dira, dirb, bfun, portd, porti, portp

The processor must be running to allow register displays. If it's running in the monitor, the emulator does the display directly. If it's running the target system program, the emulator forces a break to the monitor, gets the register data, then returns to the user program. (If you restrict the emulator to real-time runs, the **R**egister **D**isplay or **R**egister **M**odify commands aren't allowed during a user program run. See chapter 3, "Configure the Emulator.")

You can combine displays of multiple registers and register classes by listing all the arguments on the same command line.

Examples

To display the processor's X register, select:

```
Register Display Single x <Enter>
```

To display the UART registers, select:

```
Register Display Class uart <Enter>
```

To modify registers

1 Select Register Modify.

2 Use the **Tab** key to select a register name.

or

Type in the register name.

3 Press **<Enter>** to accept the name and move to the next field.

4 Type an integer or integer expression to which the register should be modified. (You cannot use symbols in these expressions.)

5 Press **<Enter>** to accept the value and modify the register.

Modifying a register's contents can help you test the effects of different program values without the trouble of rebuilding your program code. For example, you might stop the processor at a certain point (use a software breakpoint), then modify a register, and run from that point to test the result.

The processor must be running to allow modifying registers. See "To display registers" above for more information.

Examples

To modify the PC register to f0ac, select:

```
Register Modify pc <Enter> 0f0ac <Enter>
```

Using Software Breakpoints

Software breakpoints give you a way to stop program execution when the processor executes a certain instruction. Suppose that you find a problem that you believe occurs before the execution of a certain instruction. You can set a breakpoint at that location. When the processor reaches that location, the emulator will force a break to the monitor. You can then display registers or memory to determine the cause of the problem.

You can insert, enable, disable or remove software breakpoints. When you insert and enable a breakpoint, the emulator replaces the instruction at the target location with a JSRP instruction. The JSRP vector fetch causes a monitor break. The emulator replaces the JSRP instruction with the original instruction when the breakpoint is hit, or when you disable or remove the breakpoint.

Before you use breakpoints, you must choose the breakpoint vector. You will normally do this only once for every emulation session, no matter how many breakpoints you use. See chapter 3, "Configure the Emulator."

You should only set a breakpoint at the first byte of a program instruction.

You can perform a single operation on a set of breakpoints by specifying all the breakpoint addresses for that operation on the command line.

To add software breakpoints

1 Select **Breakpoints Add**.

2 Type in a single address to add one breakpoint.

or

Type in several addresses, separated by semicolons, to add several breakpoints.

3 Press **<Enter>** to save the address selection.

When you set a software breakpoint, the emulator replaces one to six bytes of code at the location you specified with a JSRP instruction, followed by zero to five NOPs. It also makes an entry in the system breakpoint table. When the processor executes the breakpoint code, the breakpoint is disabled, and the original code at those locations is replaced. The software breakpoint table entry is marked as "disabled."

When the emulator is reset, you can only add breakpoints to emulation memory. The emulator uses the monitor to insert breakpoints in target memory or on-chip RAM. If the emulator is reset, use the Processor Break command to begin running in the monitor.

Examples

To add a breakpoint at the instruction label CMD_A, select:

```
Breakpoints Add int_serv.asm:CMD_A <Enter>
```

To add breakpoints at INTERPRET_CMD and MSG_LOOP, select:

```
Breakpoints Add int_serv.asm:INTERPRET_CMD;MSG_LOOP
<Enter>
```

The **int_serv.asm:** local symbol module specification sets the default local symbol module. Therefore, you don't need to specify the local symbol module with MSG_LOOP.

To set software breakpoints

- Select Breakpoints Set Single. Then type a single address to enable one breakpoint, or type in several addresses, separated by semicolons, to enable several breakpoints. Press **<Enter>** to save your selection.

  or

- Select Breakpoints Set All to enable all currently defined breakpoints.

When a breakpoint is hit, it is disabled. If you want to reenable the breakpoint, you use the Set option to the Breakpoints command. The emulator will search the breakpoint table for the address you specify. If there is a breakpoint entry for that location, the entry is marked as "enabled" and the JSRP instruction is written to memory at that location.

When the emulator is reset, you can only set breakpoints in emulation memory . The emulator uses the monitor to set breakpoints in target memory or on-chip RAM. If the emulator is reset, use the Processor Break command to begin running in the monitor.

Examples

To enable all existing breakpoints, select:

```
Breakpoints Set All
```

To enable existing breakpoints at INTERPRET_CMD and MSG_LOOP, select:

```
Breakpoints Set Single
int_serv.asm:INTERPRET_CMD;MSG_LOOP <Enter>
```

To clear software breakpoints

- Select Breakpoints Clear Single. Then type a single address to disable one breakpoint, or type in several addresses, separated by semicolons, to disable several breakpoints. Press **<Enter>** to save your selection.

  or

- Select Breakpoints Clear All to disable all currently defined breakpoints.

Sometimes you will want to temporarily disable a software breakpoint without removing it. The Breakpoints Clear command lets you do this.

When you disable a software breakpoint, the emulator replaces the JSRP instruction at the breakpoint location with the original instruction. It marks

the software breakpoint table entry as "disabled." Then the processor won't break to monitor when the instruction at that location is executed.

When the emulator is reset, you can only clear breakpoints in emulation memory. The emulator uses the monitor to clear breakpoints in target memory or on-chip RAM. If the emulator is reset, use the Processor Break command to begin running in the monitor.

Examples

To disable all existing breakpoints, select:

```
Breakpoints Clear All
```

To disable existing breakpoints at INTERPRET_CMD and MSG_LOOP, select:

```
Breakpoints Clear Single
int_serv.asm:INTERPRET_CMD;MSG_LOOP <Enter>
```

To remove software breakpoints

- Select **B**reakpoints **R**emove **S**ingle. Then type a single address to remove one breakpoint, or type in several addresses, separated by semicolons, to remove several breakpoints. Press **<Enter>** to save your selection.

  or

- Select **B**reakpoints **R**emove **A**ll to remove all currently defined breakpoints.

When you're finished using a particular breakpoint, you should remove the breakpoint table entry. The **B**reakpoints **R**emove command lets you do this. The original instruction is restored to memory, and the breakpoint table entry is removed.

When the emulator is reset, you can only remove breakpoints from emulation memory. The emulator uses the monitor to remove breakpoints from target

memory or on-chip RAM. If the emulator is reset, use the Processor Break command to begin running in the monitor.

Examples

To remove all existing breakpoints, select:

```
Breakpoints Remove All
```

To remove existing breakpoints at INTERPRET_CMD and MSG_LOOP, select:

```
Breakpoints Remove Single
int_serv.asm:INTERPRET_CMD;MSG_LOOP <Enter>
```

To display software breakpoints

- Select Breakpoints Display.

The status of each breakpoint is shown.

Making Coordinated Measurements

The Coordinated Measurement Bus (CMB) connects multiple emulators and allows you to make synchronous measurements between those emulators.

For example, you might have a target system that contains an HPC processor and another processor. You use HP 64700 Series emulators to replace both target system processors, and connect the emulators using the CMB. Then, you might want to start a program run on both emulators simultaneously. Or, you may want to start a trace on one emulation analyzer when the other emulator reaches a certain program address. These measurements are possible with the CMB.

To connect emulators using the CMB, see the *Installation/Service Guide.*

To start two emulators simultaneously

1 Set the run address for the first emulator by selecting Processor Cmb Go Address **<address> <Enter>**.

2 Set the run address for the second emulator by selecting Processor Cmb Go Address **<address> <Enter>**.

3 Start program execution on both emulators by selecting Processor Cmb Execute.

Before you do this procedure, both emulators must be connected via the CMB. Also, they should be configured, and either reset, or running in the monitor.

Using the Emulator In-Circuit

Out-of-circuit emulation is useful for debugging your program code. You can use the emulation analyzer and other emulator features to test and evaluate your code.

As your target system design progresses, you'll want to test your program on the system hardware.

You connect the emulator probe to your target system to do in-circuit emulation. Then you can make analyzer measurements and have the memory display and other capabilities of the emulator to integrate your program code with the target system.

To prepare for in-circuit emulation

1 Study your target system design, especially the memory configuration and bus width.

2 Study chapter 4 of this manual to determine how to set configuration items for the best results with your target system.

3 Install the target system probe.

4 Set configuration items as determined by the results of step 2.

When you use the emulator in-circuit, you need to carefully consider the configuration of the emulator and its relationship to your target system design. See the chapter "Configure the Emulator" for details.

To install the target system probe

Caution

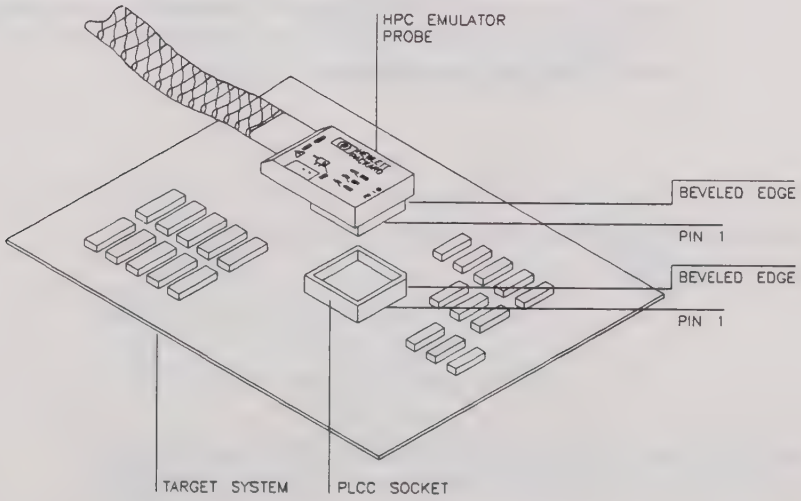
Possible damage to the emulator probe. The emulation probe contains devices that are susceptible to damage by static discharge. You should take precautions before handling the probe, to avoid damaging the internal components of the probe with static electricity.

Caution

Possible damage to the emulator. Make sure both target system and emulator power are OFF before installing the emulator probe into the target system.

Caution

The emulator probe will be damaged if incorrectly installed. Make sure to align pin 1 of the probe connector with pin 1 of the socket.



HPC Emulator Probe Installation

1 Remove the HPC processor from the target system socket. Note the location of pin 1 on the processor and on the target system socket. Store the processor in a protected environment (such as antistatic foam).

2 Insert the emulator probe into the target system socket. Make sure to align pin 1 of the emulator probe and the target system socket.

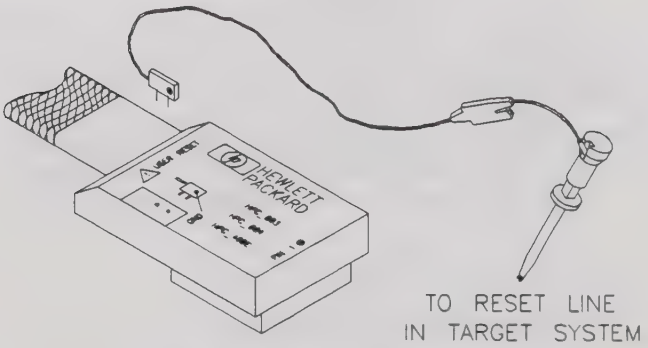
To connect emulator reset to target system reset

1 Power off the target system and the emulator.

2 Connect the two-pin reset lead to the emulator probe.

3 Connect the reset lead grabber probe to the target system RESET line.

The HPC processor reads the state of the target system HBE and EXM lines when it is reset from the target system. It uses these lines to determine the correct chip mode and bus configuration. You should connect the emulator reset to target system reset to ensure that these lines are read and configured whenever you reset the emulator.

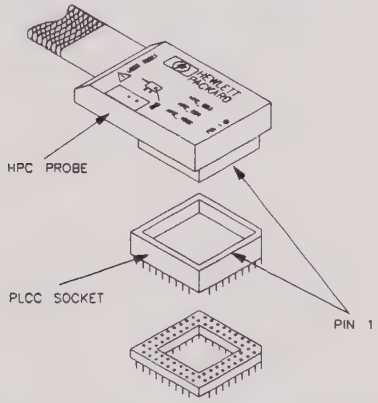


Connecting Emulator Reset to Target System Reset

To install the PLCC to PGA adapter

- Install the PLCC to PGA adapter as shown in the figure below.

  If necessary, you can stack additional female-to-male pin protectors on the probe for added height.



PLCC to PGA Adapter Installation

To power-on the emulator and target system

1 Apply power to the emulator.

2 Apply power to the target system.

Caution

Possible damage to the emulator! You must apply power to the emulator before you apply power to the target system. Otherwise, the emulator may be damaged.

3

Using the Analyzer

How to record program execution in real-time

The emulation analyzer is a powerful tool that allows you to view the execution of your program in real-time. Powerful triggering and sequencing capability ensures that the analyzer captures only the information you need, so you don't spend time searching through detailed trace lists for the information that's of interest.

This chapter tells you how to use some of the basic analyzer capability. More detailed instructions for the analyzer can be found in the *Analyzer Terminal Interface User's Guide.*

If you don't have the external analyzer, simply omit the Internal option from the commands shown in this chapter.

Making Basic Analyzer Measurements

You can use just a few analyzer commands to make most measurements, such as these:

- Start or stop a trace measurement.
- Display the trace status.
- Display the trace list.
- Define a simple trigger qualifier.
- Define a simple storage qualifier.
- Set the trigger position in trace memory.

The analyzer has powerful triggering, storage and trace list display capability. These features are described in other sections of this chapter.

To start a trace measurement

- To begin an emulation analyzer trace, select: **Analysis Begin Internal**

When you start a trace, the analyzer begins recording data according to your trigger and storage specifications. When the trace is complete, or halted, you can display the data.

To halt a trace measurement

- To halt an emulation analyzer measurement, select: **Analysis Halt Internal**

Sometimes you need to halt a trace because an examination of the analyzer status shows that the trace isn't capturing the data you expect. Then, you'll want to halt the analyzer and reconfigure your trigger and storage terms to capture data.

To display the trace status

- To see the emulation analyzer trace status, look at the PC Interface status line.

The trace status display shows whether the trace is running, halted or complete.

To display the trace list

1 To display the trace list using the default parameters, select: Analysis Display Internal

2 Then press **<End><Enter>** to accept the default display.

The trace list is a 512 or 1024 state deep buffer. You can selectively display portions of the buffer using the Analysis Display form. See the *Analyzer PC Interface User's Guide* for more information.

At the beginning of each trace list, you may see some accesses to the HPC stack and vector area that are not part of your program. These accesses are background monitor accesses to the stack and accesses to the vector table to define the software breakpoint vector and reset vector.

To define a simple trigger qualifier

1 To define a simple trigger on an address value, select: Analysis Trace Modify Internal

2 Use the arrow keys to move to the **Trigger on** field and type **a**. Press **<Enter>**.

3 Use the arrow keys to move to the address field next to **a=**. Type in the address value. Press **<End><Enter>** twice to save the trigger specification.

Many times, you'll want to trigger the analyzer (begin storing data) when a certain program location is reached. You can use either a simple address expression, or one that includes symbols.

Example

To trigger the analyzer when the demo program reaches the location FILL_DEST, enter:

```
a=int_serv.asm:FILL_DEST
```

Where int_serv.asm:FILL_DEST is positioned under the address column.

To define a storage qualifier

1 To have the analyzer store only cycles that reference a particular address range, select: **A**nalysis **T**race **M**odify **I**nternal

2 Use the arrow keys to move to the Store field. Type **r** and press **<Enter>**.

3 Use the arrow keys to move to the field next to the **Range** label. (Leave the label as addr). Type in the lower address, press **<Enter>**, and type in the upper address.

4 Press **<End><Enter>** twice to save the new trace specification.

If you want to store only the accesses to a certain range of locations, you can use the trace storage qualifier. More complex patterns of data and status qualification can be made. See the *Analyzer PC Interface User's Guide* for details.

Examples

If you want to measure the amount of time between interrupts for the emulator demo program, set the trigger pattern to HANDLE_CMD (as

described earlier in this chapter). Then set the storage qualifier for all levels to **b** (instead of a range) and set **b=HANDLE_CMD**.

Start the trace:

```
Analysis Begin Internal
```

Then run the program:

```
Processor Go Address INIT <Enter>
```

Display the trace:

```
Analysis Display Internal <End> <Enter>
```

The display will show that the interval between interrupts is about 16.7 ms.

To set the trigger position

1 To position the trigger term, select: **A**nalysis **T**race **M**odify **I**nternal

2 Press the **<End>** key to move to the last field in the trace specification.

3 Use the **<Tab>** key to select **start, center** or **end** to position the trigger at the start, center or end of the trace. Or type in a value from 1 through 512 (1 through 1024 if counting is off) to position the trigger at that trace state.

4 Press the **<Enter>** key to save the new trace specification.

The trigger position can help make the trace list more readable. For example, you might want to see all the program events leading to a particular access. So, you can define that access as the trigger term, then position the trigger at the end of the trace.

Examples

To position the trigger 10 states after the beginning of the trace, select:

```
Analysis Trace Modify Internal
```

Press the **<End>** key to move to the last field. Type **10** and press **<Enter>** to save the trace specification.

You'll find this useful for HPC analysis, since the HPC prefetches instructions.

Displaying the Trace List

The *trace list* is your view of the analyzer's record of processor bus activity. You can alter the information presented in the trace list to make it easier to find the information of interest. For example, you can display symbol information where available, or source lines from the high-level languages used to write the target system program. You can also change the column widths and set options for disassembly of the trace list.

To change the trace format

1 Select Analysis Format Internal.

2 Use the arrow keys to move to one of the "Label" fields.

3 **Tab** or **Shift-Tab** to change the label to one of the following:

addr (for address lines)
data (for all 16 data lines)
lowdata (low 8 data lines; byte data)
mne (for instruction mnemonics)
stat (for processor status information)
count (state and time counts)
seq (sequencer state change indicator)
xbits (external analyzer inputs)
--OFF-- (no label in this position)

Press **<Enter>** to accept the new label definition.

4 If you selected **xbits**, you can select the polarity of the inputs. **Tab** to select **+** for positive polarity or **-** for negative polarity.

5 If a **base** specification is available for the label, you can move to it with the arrow keys. Press **<Tab>** to select one of the following bases:

hex (hexadecimal)
bin (binary)
oct (octal)
dec (decimal)

Press **<Enter>** to accept the base specification.

6 For the addr label, you may specify a field width, but only when the mne label is in use. Type in a field width in the range **4..50** and press **<Enter>**.

7 For the count label, you may specify **rel** to show the count relative to the previous state or **abs** to show the count relative to the trigger state. Press **Tab** to select the desired setting and press **<Enter>**.

8 Press **<End>**, then **<Enter>** to save the format changes and exit the format screen. Press **<Esc>** to exit the format screen and discard your changes.

The Analysis Format form specifies how data is arranged on the screen when you display the trace list. You can specify multiple options on the form. The sequence of the options in the form determines the sequence of the columns in the trace list display.

When you enter a Analysis Format command with a new set of options, the previous trace format is destroyed and the options for the new command set the format (see the examples).

To display the trace list

1 Select: Analysis Display Internal.

2 The cursor is now in the "Address disassembly mode" field. **Tab** to select **address** if you want to see numeric addresses in the address field, **symbols** if you want to see only symbols in that field (when available), or **both** if you want to see a mix of the numeric addresses and symbols. Press **<Enter>**.

3 Now the cursor is in the "Start" field. Type in a number to specify the first state to be displayed from the trace list. Press **<Enter>**.

4 Now the cursor is in the "End" field. Type in a number to specify the last state to be displayed from the trace list.

5 Press **<Enter>** to accept your choices and display the trace list. (If you did not move to the last field, press **<End>** before pressing **<Enter>**.) Press **<Esc>** to abort the Analysis Display command.

The trace list is a 512 or 1024 state deep buffer. You can selectively display portions of the buffer using the Analysis Display form.

To change the trace depth

1 Select Analysis Trace Modify Internal.

2 Use the arrow keys to move the cursor the field next to "Count" and press the **Tab** key to select **off** to disable counting. Press **<Enter>**.

3 Press **<End>**, then **<Enter>** to save your changes and exit the trace specification form. Press **<Esc>** if you want to discard your changes and exit the trace specification form.

The analyzer's state/time counter uses half of the analyzer's state memory resources. Therefore, when you use the time or state counter, the analyzer's trace depth is 512 states. You can double the trace depth to 1024 states by setting the state/time counter to off.

Making Complex Measurements

You can have the analyzer record bus activity by simply using the trace command without any options. But this doesn't use the analyzer effectively for two reasons:

- the trace memory may fill before the program reaches the states of interest;
- you must search through the trace list to find a few states pertinent to your measurement problem.

The HP 64700 analyzer has trigger and sequence capabilities that help solve these problems. These tools act as a filter for processor bus activity that allows the analyzer to capture only the states you want to see in the measurement.

A *trigger* tells the analyzer to start storing data when a certain bus state is encountered. A *sequence* is a more complex specification that specifies a series of bus states that must be satisfied before the trigger is reached.

This section tells you how to get the most from the HP 64700 analyzer by specifying triggers and sequence specifications. It also describes additional measurement tools to help you get more information from the trace.

Expressions are an important part of trace specifications, since they specify the numeric or logical values that the analyzer matches for trigger and storage. Expressions are represented by the <expression> symbol in this chapter.

To insert a sequence term

1 Select **Analysis Trace Modify Internal**.

2 Use the arrow keys to move the cursor to the term number of the sequencer term where you want to insert a new term. The new term will be inserted before that term.

3 **Tab** until the term number field displays **I**, and press **<Enter>**. A new sequence term is inserted.

4 Make any changes needed to qualifiers (see other parts of this chapter for details.)

5 Press **<End>**, then **<Enter>**, to save your changes and exit the trace specification form. Press **<Esc>** if you want to discard your changes and exit the trace specification form.

You can specify up to eight sequence terms. At least two are always required, because the first term specifies the trigger and the analyzer triggers on entry into the succeeding term.

When you start a trace, the analyzer searches for the "Find" state of the first sequence term, then searches for the "Then find" state of the second term, and so on until it reaches the "Trigger on" term.

Any one of the eight sequence terms may be specified as the trigger level. The trigger state is always line 0 in the resultant trace list.

When you insert a new sequence term in the middle of the sequence, succeding terms are incremented.

To remove a sequence term

1 Select **A**nalysis **T**race **M**odify **I**nternal.

2 Use the arrow keys to move the cursor to the term number of the sequencer term that you want to remove.

3 **Tab** until the term number field displays **D**, and press **<Enter>**. The sequence term is deleted.

4 Make any changes needed to qualifiers (see other parts of this chapter for details.)

5 Press **<End>**, then **<Enter>**, to save your changes and exit the trace specification form. Press **<Esc>** if you want to discard your changes and exit the trace specification form.

You may want to delete sequence terms to remove unneeded qualifications from the sequence specification. When you delete a sequence term, any terms above it are decremented to fill the gap.

To assign the trigger term

1 Select Analysis Trace Modify Internal.

2 Use the arrow keys to move the cursor to the term number of the sequencer term that you want to use as the trigger level.

3 **Tab** until the term number field displays T, and press **<Enter>**. The "Find" or "Then find" label changes to "Trigger on."

4 Make any changes needed to qualifiers (see other parts of this chapter for details.)

5 Press **<End>**, then **<Enter>**, to save your changes and exit the trace specification form. Press **<Esc>** if you want to discard your changes and exit the trace specification form.

Any but the last of the eight sequence terms may be specified as the trigger level. The trigger state is always line 0 in the resultant trace list.

The selected term specifies the trigger qualifier. The analyzer triggers on entry into the succeeding term. (See the description of complex configuration in the *Terminal Interface User's Guide* for more information.)

To reset the analyzer

- Select Analysis Trace Reset Internal.

When you reset the analyzer trace specification, the sequencer is set to the default two-term sequence in which it stores any state and triggers on any state. All pattern and range specifications are also reset. (See "To define trace patterns" and "To define a range" for more information.)

To define a qualifier

1 Select Analysis Trace Modify Internal.

2 Use the arrow keys to move to the qualifier field for which you want to specify an expression. (The analyzer searches for the expression you specify before taking action, such as counting a state, moving to a new sequence term, and so on.)

3 Tab to select a state qualifier from the following list of states, ranges and patterns:

any state, no state, r, !r, a, b, c, d, e, f, g, h, arm

or

Type in a complex expression using patterns and ranges according to the following rules:

- The patterns, range and arm qualifier are divided into two disjoint sets.

  <SET1> = {a,b,c,d,r,!r}
  <SET2> = {e,f,g,h,arm}

  (The **arm** qualifier is discussed in the section on coordinated measurements.)

- You can form expressions by inserting intraset operators between members of the same set. The operators are:

  ~ (intraset logical NOR)
  | (intraset logical OR)

  If you form an expression using these operators, the operator must remain the same for all members of the same set. (See the examples).

- You can form expressions by inserting interset operators between members of **<SET1>** and **<SET2>**. The operators are:

  and (logical and)
  or (logical or)

  The order in which you put the sets does not matter.

4 Press **<Enter>** if you need to define the expressions associated with the patterns and ranges in the expression. (See "To define trace patterns.") Otherwise, skip to step 5.

5 Press **<End>**, then **<Enter>**, to save your changes and exit the trace specification. Press **<Esc>** to discard your changes and exit the trace specification.

Complex expressions allow you to build more complicated trace qualifiers with multiple conditions. Since the complex expressions are built from the trace patterns, which contain simple expressions, you can build qualifiers with multiple logical operators.

To create an expression

- Form logical expressions by combining numeric values and logical operators to produce a numeric result.

The simplest numeric expressions consist of numbers and radix indicators. The radix indicators are:

Y y (binary)

Q q O o (octal)

T t (decimal)

H h (hexadecimal (default))

See the *PC Interface Reference* for more details on numeric expressions and the available logic operators.

Examples

The following are valid numeric expressions:

```
1XXX0Y<<3

(340q*7)/2

0ffa^32T

52T*7a
```

The PC Interface defaults to hexadecimal radix; therefore, you don't need to specify 7ah in the last example.

To define trace patterns

1 Select Analysis Trace Modify Internal.

2 Use the arrow keys to move to the qualifier you want to change and enter a pattern, range or complex expression. (See "To define a qualifier" for more details.) Press **<Enter>**.

3 Now you are in the pattern modification screen. Use the arrow keys to move to the equality field next to the desired pattern. To set the pattern equal to an expression, Tab until this field displays **=**. To set the pattern not equal to the expression, Tab until the field displays **!=**. Press **<Enter>** to move to the addr field for that pattern.

4 In the addr field, you can enter an address expression, using either numeric expressions, symbols, or a combination thereof. Or, you can leave the field blank. Press **<Enter>** to move to the data field.

5 In the data and lowdata fields, you can enter data expressions using numeric expressions, or leave them blank. Use **data** to match 16-bit data patterns. Use **lowdata** to match data for byte operations. Press **<Enter>** to move to the stat field.

6 In the stat field, you can enter a status expression using numeric values. Or, you can use the **Tab** key to select from different predefined status qualifiers. (See chapter 9 for a list.) You can use logical operators to combine the status qualifiers.

7 If the external analyzer is present and aligned with the internal analyzer, you can enter an expression in the **xbits** field. Or, you can leave the field blank. Press **<Enter>** to move to the next field.

8 When you finish modifying patterns, press **<End>** to move to the last field in the form. Press **<Enter>** to accept your changes and return to the sequencer definition screen. You can exit that screen by pressing **<End>**, then **<Enter>**.

If you want to discard your changes, press **<Esc>**.

The analyzer provides eight pattern variables to which you assign simple expressions. Then you use these patterns to build more complicated qualifiers for the primary and secondary branch qualifiers.

By specifying the address, data and status values for a particular pattern, you can use a pattern to specify a complete processor bus state.

The equality setting affects the logical connectives between the address, data and status values. If you set the equality to **=**, the logical connective is **and**. If you set the equality to **!=**, the logical connective is **or**. For example, you could have the following settings for pattern a:

```
addr=CMD_INPUT and data=41

addr!=CMD_INPUT or data!=41
```

You can combine the predefined status qualifiers. For example, if you want to see only foreground read cycles, you would enter the following in the stat field:

```
read&&foreground
```

To define a range

1 Select Analysis Trace Modify Internal.

2 Use the arrow keys to move to the qualifier you want to change and enter an expression using the range expression (**r** or **!r**). (See "To define a qualifier" for more details.) Press **<Enter>**.

3 Now you are in the pattern modification screen. Use the arrow keys to move to the label field next to "Range (r) Label." **Tab** to select **addr** (address) or **data** (data) as the group of processor lines for which you want to define a range. Press **<Enter>** to move to the next field.

4 Now you can specify the lower bound of the range. If you selected addr, you can enter an address expression, using either numeric expressions, symbols, or a combination thereof. If you selected data, you can enter data expressions using numeric expressions. Press **<Enter>**.

5 Specify the upper bound of the range, using the same rules as in step 5.

6 When you finish, press **<End>** to move to the last field in the form. Press **<Enter>** to accept your changes and return to the sequencer definition screen. You can exit that screen by pressing **<End>**, then **<Enter>**.

If you want to discard your changes, press **<Esc>**.

The range qualifier **r** can be used in analyzer storage and complex branch qualifiers. For example, you might have a lookup table in your program, and want to record accesses to that table in the trace list. You can define the range qualifier as the set from the lower to upper boundaries of the lookup table.

To set the storage qualifier

1 Select Analysis Trace Modify Internal.

2 Enter a qualifier in the qualifier field next to the "While Storing" or "Store" label for each sequence term. (See "To define a qualifier" if you need more information.)

3 Press **<End>**, then **<Enter>**, to save your changes and exit the trace specification form. Press **<Esc>** if you want to discard your changes and exit the trace specification form.

There are eight storage qualifiers, one for each sequencer term. This allows you to store only the states of interest at each level of the sequence, which uses the trace memory more efficiently and makes the trace display easier to read.

To define a primary branch term

1 Select Analysis Trace Modify Internal.

2 For each term that you want to modify, enter a state qualifier in the qualifier field next to the "Find" or "Then find" labels. (See "To define a qualifier" if you need more information.) Press **<Enter>** if you need to define patterns, or use the arrow keys to move to the "times" field.

3 Now the cursor is in the field next to the word "times." This field sets the occurrence count. Type in a number between 1 and 65535 that specifies how many times the qualifier must be found before the sequencer advances to the next state.

4 Press **<End>**, then **<Enter>**, to save your changes and exit the trace specification form. Press **<Esc>** if you want to discard your changes and exit the trace specification form.

The primary branch qualifier defines the main path from a given sequencer term to another term (or the same term). If both the primary and secondary branch qualifiers are satisfied simultaneously, the primary branch is taken.

Usually, you'll use the primary branch qualifiers to define a sequence of states that must be satisfied to reach the trigger condition.

To define a global restart term

1 Select Analysis Trace Modify Internal.

2 Use the arrow keys to move the cursor to the field next to the label "Branches." **Tab** to select **restart on** if you want a global restart, or **off** if you want to disable the global restart. Press **<Enter>**.

3 If you selected **restart on** in step 2, enter a state qualifier in the qualifier field below the "Branches" field. (See "To define a qualifier" if you need more information.)

4 Press **<End>**, then **<Enter>**, to save your changes and exit the trace specification form. Press **<Esc>** if you want to discard your changes and exit the trace specification form.

The restart qualifier allows you to restart the trace sequence whenever a certain instruction or data access occurs. For example, you might have a complicated trace sequence that searches for an intermittent failure condition. You could set the restart term to restart the sequence whenever a bus cycle occurred that ensured that the code segment would perform correctly. Thus, the trace will be satisified only when that restart term never occurs and the code segment fails.

You can also have the analyzer take a unique secondary branch for each sequencer level. See "To define a secondary branch term."

To define a secondary branch term

1 Select Analysis Trace Modify Internal.

2 Use the arrow keys to move the cursor the field next to the label "Branches." **Tab** to select **per level** if you want to specify individual secondary branch qualifiers, or **off** if you want to disable the secondary branch qualifiers. Press **<Enter>**.

3 If you selected **per level** in step 2, enter a state qualifier in the qualifier field next to the "Else on" label for each sequence term. (See "To define a qualifier" if you need more information.) Press **<Enter>** to define patterns or use the arrow keys to move to the "goto level" field. Type in a sequence level (only the sequence levels currently displayed are valid).

4 Press **<End>**, then **<Enter>**, to save your changes and exit the trace specification form. Press **<Esc>** if you want to discard your changes and exit the trace specification form.

The secondary branch qualifier defines an alternate path from a given sequencer term to another term (or the same term). If both the primary and secondary branch qualifiers are satisfied simultaneously, the primary branch is taken. You can use the secondary branch as an alternate path to the trigger if more than one sequence of conditions is acceptable.

Since the secondary branch qualifier is unique for each sequence term, it is more flexible than the global restart qualifier. You can make all the secondary branch qualifiers identical. This is called global restart. See "To specify a global restart qualifier."

Using Other Analyzer Features

The analyzer has other features that can be used in all configurations to make trace measurements easier to interpret or add additional information.

- Prestore allows you to save specific trace states that are related to other events in your trace list. For example, you might want to save the caller of a subroutine.
- Count qualifiers allow you to count states or time.

To define a prestore qualifier

1 Select Analysis Trace Modify Internal.

2 Use the arrow keys to move the cursor the field next to "Prestore" and press the **Tab** key to select **on**. (Select **off** if you want to disable a previous prestore specification, and skip to step 4.) Press **<Enter>**.

3 Type in a state qualifier expression in the field below "Prestore." (See "To define a qualifier" in this chapter.)

4 Press **<End>**, then **<Enter>** to save your changes and exit the trace specification form. Press **<Esc>** if you want to discard your changes and exit the trace specification form.

You use the prestore qualifier to save states that are related to other routines that you're tracing. For example, you might be tracing a subprogram, and want to see which program called it. You specify the caller as a prestore specification.

To count states or time

1 Select Analysis Trace Modify Internal.

2 Use the arrow keys to move the cursor the field next to "Count" and press the **Tab** key to select **time** if you want to count time intervals, **state** if you want to count bus states, or **off** if you want to disable counting. Press **<Enter>**.

3 If you selected **state** in step 2, type in a state qualifier expression in the field below "Count." (See "To define a qualifier" in this chapter.)

4 Press **<End>**, then **<Enter>** to save your changes and exit the trace specification form. Press **<Esc>** if you want to discard your changes and exit the trace specification form.

The trace count qualifier can be used to measure time for each storage state or occurrence counts of a particular bus state. You can display these values either relative to the last stored state (relative mode) or relate to the trigger state (absolute mode). You change the display using the Analysis Format command. See "To change the trace format" earlier in this chapter.

When you select **off**, the analyzer trace depth is increased from 512 states to 1024 states.

4

Configuring the Emulator

How to adapt the emulator to your target system design

Each HPC target system differs in the way it configures the HPC processor, uses microcontroller resources, uses memory and memory mapped I/O devices. During system development, your needs for emulator resources may change as your target system design matures. You can allocate HPC emulator resources by filling in PC Interface configuration forms. This resource allocation is called the emulator configuration.

Target Interface

You can select the clock source for the processor. Also, you can enable or disable a target system reset signal.

Interrupt Handling

The HPC emulator uses the non-maskable interrupt as a hardware break signal. If your target system uses this interrupt, you can control sharing of the interrupt between the emulator and your system.

Memory

The emulator must know how your target system memory resources are allocated. You can use emulation memory for some memory ranges. This is useful in the early stages of target system design, and for emulating on-chip memory.

A bank switched mode allows use of a 20-bit address for both target and emulation memory. The range of memory that is bank switched can be configured.

A special zero wait state mode allows you to develop applications using emulation memory for systems that won't have off-chip memory.

Emulation Monitor

The emulation monitor is used to implement some emulator features. For example, display or modification of target system or on-chip memory is done by the monitor. You can choose either a foreground or background monitor.

You can enable interrupts during foreground monitor functions. User interrupt service routines can interrupt this monitor to preserve real time response to target events.

You can specify custom subroutines that are executed when certain monitor events occur. These subroutines will work with either the foreground or background monitor.

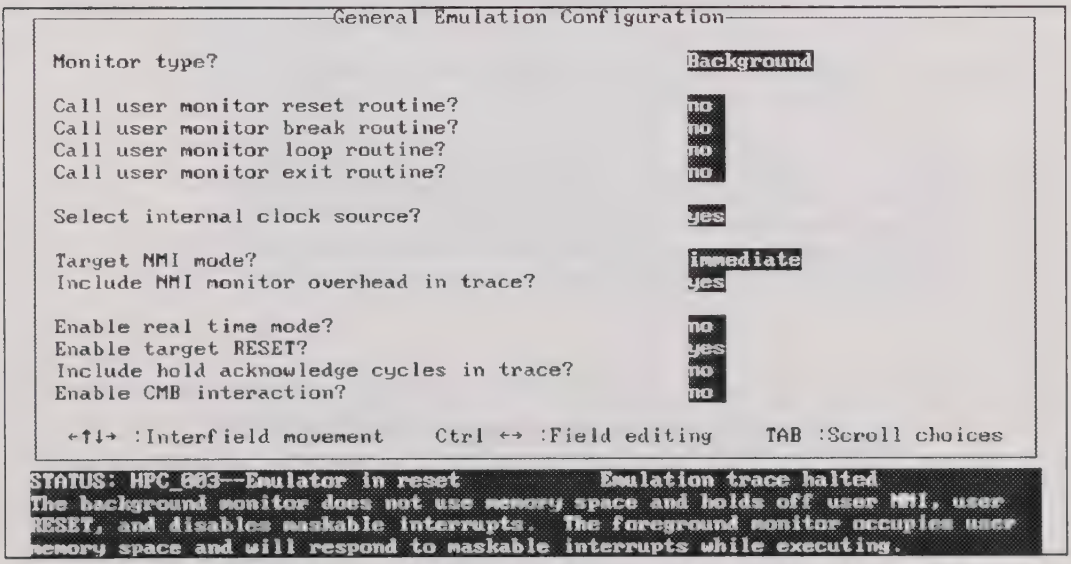
Break Conditions

Software and hardware breakpoints allow you to terminate your program and start the monitor.

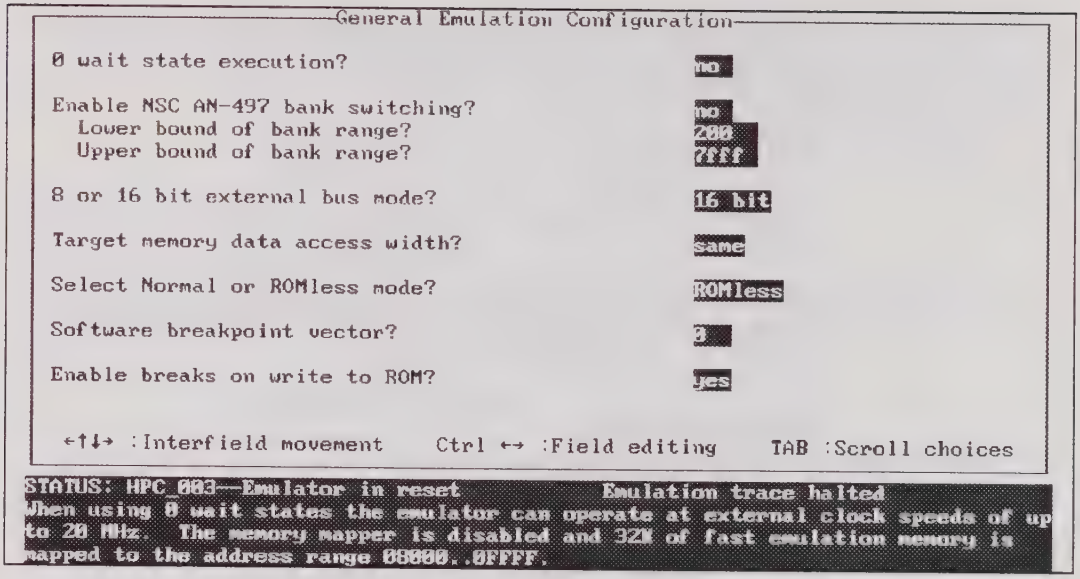
The software breakpoint features requires the use of one of the 16 HPC subroutine vectors. You can choose which vector is used, or prevent use of a vector if your application needs all the vectors.

Analyzer Interface

You can enable the analyzer to recognize some types of processor activity that is usually ignored. For example, you can control how emulation NMI interrupt arbitration is displayed in the trace list. You also can enable the analyzer to report the occurrence of target system DMA activity.


General Emulation Configuration
Monitor type? Background
Call user monitor reset routine? no
Call user monitor break routine? no
Call user monitor loop routine? no
Call user monitor exit routine? no
Select internal clock source? yes
Target NMI mode? immediate
Include NMI monitor overhead in trace? yes
Enable real time mode? no
Enable target RESET? yes
Include hold acknowledge cycles in trace? no
Enable CMB interaction? no
←↑↓→ :Interfield movement Ctrl ↔ :Field editing TAB :Scroll choices
STATUS: HPC_003--Emulator in reset Emulation trace halted
The background monitor does not use memory space and holds off user NMI, user RESET, and disables maskable interrupts. The foreground monitor occupies user memory space and will respond to maskable interrupts while executing.

Config Emulator Form


General Emulation Configuration
0 wait state execution?
no
Enable NSC AN-497 bank switching?
no
Lower bound of bank range?
2000
Upper bound of bank range?
2fff
8 or 16 bit external bus mode?
16 bit
Target memory data access width?
same
Select Normal or ROMless mode?
ROMless
Software breakpoint vector?
3
Enable breaks on write to ROM?
yes
←↑↓→ :Interfield movement
Ctrl ↔ :Field editing
TAB :Scroll choices
STATUS: HPC_003--Emulator in reset
Emulation trace halted
When using 0 wait states the emulator can operate at external clock speeds of up to 20 MHz. The memory mapper is disabled and 32K of fast emulation memory is mapped to the address range 00000..0FFFF.

Config Pod Screen Form

Configuring the Target System Interface

The target system interface configuration determines:

- HPC processor clock source.
- Target system reset signal acceptance.

To select the processor clock source

1 Select Config Emulator.

2 Use the arrow keys to move to the field next to "Select internal clock source?"

3 **Tab** to select **yes** if you want to use the emulator's internal 20 MHz clock. Select **no** if you want to use the target system clock.

4 Press **<End>**, then **<Enter>** to save your changes and exit the configuration. Press **<Esc>** to discard your changes and exit the configuration.

When you change the clock source, the emulator is reset. You can't select the external clock source if the emulator is not connected to a powered-on target system. When you connect the emulator to the target system, you can use the internal clock, but you'll eventually want to use the external clock as your system design matures. The internal clock (yes) is the default setting.

To enable or disable target system resets

1 Select Config Emulator.

2 Use the arrow keys to move to the field next to the question "Enable target RESET?"

3 **Tab** to select **yes** if you want to allow target system resets to reset the processor. Select **no** if you want to prevent the target system from resetting the processor.

4 Press **<End>**, then **<Enter>** to save your changes. Press **<Esc>** to discard your changes.

When you disable target system resets, the processor ignores all target system resets unless you select Processor Go Reset. Then the target system reset is enabled so that the processor will run when that reset occurs.

You might disable resets if your target system reset circuitry is incomplete, or isn't functioning correctly.

When you enable target system resets, the processor always receives the reset, unless:

- The processor is executing in the background monitor,

  or

- The processor is executing a user NMI service routine when the NMI mode configuration item is set to **immediate** or **wait**.

In these cases, the emulator will hold off the user reset signal until the condition terminates. The emulator will always generate a signal of the correct duration, even if the hold-off shortened the user reset signal.

When you use the emulator in-circuit, you will usually want to enable target system resets. If you've connected the watchdog output to the target system reset line (to trap infinite loops or errant code), you should enable resets to ensure that the HPC is reset when the WO signal is asserted.

To configure non-maskable interrupt sharing

1 Select Config Emulator.

2 Use the arrow keys to move to the field next to the question "Target NMI mode?"

3 **Tab** to choose one of four modes as follows:

To share the NMI with the target system, and execute emulator break requests immediately, choose **immediate**.

To share the NMI with the target system, and wait for user NMI routines to complete before a break is honored, choose **wait**.

To share the NMI with the target system, without regard for the effect of nested user NMI's, choose **fast**.

To disable the target system NMI, choose **disable**.

4 Press **<End>**, then **<Enter>** to save your changes. Press **<Esc>** to discard your changes.

When you change the setting, the processor is reset. The default setting is **immediate**.

When you choose **immediate**, **wait** or **fast** modes, the emulator accepts both target system NMI's and those from the emulator (the hardware break). Your programs must initialize the NMI vector and supply an interrupt service routine for the target system NMI.

If the emulator is executing in the background monitor, user NMIs are held off by the emulator until the monitor exits. Then the user NMI will be asserted. If you're using the foreground monitor, the user NMI is accepted when the foreground part of the monitor is entered.

When an NMI occurs in **immediate** or **wait** modes, some background monitor code is executed to determine the source of the NMI. If it's a user NMI, the monitor saves the Global Interrupt Enable (GIE) bit state. When a user NMI routine exits, the emulator enters the monitor to restore the previous GIE state. This ensures that nested user NMI's don't destroy the GIE state. (Nested NMI's can occur when the emulator asserts a break request during a user NMI service routine.)

The overhead for saving and restoring the GIE state is as follows:

Processor Clock Speed	at 30 MHz, 1 wait state	at 20 MHz, 0 wait states
Entry overhead (save the GIE bit)	30 μs	25 μs
Exit overhead (restore the GIE bit)	25 μs	20 μs

If you want emulator breaks to occur without waiting for a target system NMI service routine to complete, select **immediate**. The target NMI service routine will resume when the monitor exits, unless you modify the PC value while in the monitor.

If you want to delay an emulator break until the target system NMI service routine is finished, select **wait**. You might choose this mode if the user NMI service routine is very short, and you don't want it to be interrupted.

Certain monitor events are never delayed, even in **wait** mode. There are: software breakpoints, step, return from user monitor routine, return from a foreground monitor, guarded memory access breaks and write to ROM breaks.

When an NMI occurs in **fast** mode, the user NMI service routine is executed immediately. The monitor is never entered, and there is no overhead added to the NMI service routine.

Don't use **fast** mode if an emulator break could interrupt a user NMI service routine in an application that enables maskable interrupts. The break will (incorrectly) clear the CGIE bit in the PSW register, which will disable maskable interrupts.

Configuring and Mapping Memory

Each target system allocates memory and I/O as needed by the application. As the system design matures, memory locations and requirements may change. For example, the initial target system design may not support external memory. But a change in application definition needs more program code, requiring external memory. While the design is being changed, you can develop the program using the HPC emulator's emulation memory to simulate target system memory.

The HPC emulator has flexible memory resources that allow you to configure the emulator to support your needs. The memory configuration items are as follows:

Access Mode

Determines the way in which the emulator accesses target system memory.

Bank Switching

Select whether bank switching is used, and define the address range used in banks.

Memory Allocation

Define address ranges as being emulation or target system memory. Label these ranges as ROM or RAM. Select fast emulation of on-chip memory while in-circuit.

Since the memory map is reset whenever you change the chip, wait state, or bank configuration, or change the monitor, you should set these configuration items before assigning emulation and target system memory map terms.

To set the access mode

1 Select Config Pod.

2 Use the arrow keys to move the cursor to the field next to the question "Target memory data access width?"

3 **Tab** to select **byte**, **word** or **same**, which do the following:

Width Setting	Action
byte	The emulation monitor uses byte instructions to access target system memory.
word	The emulation monitor uses word instructions to access target system memory.
same	The emulation monitor uses instruction types as determined by the operation and emulation configuration. If you select the 8 bit bus width in the configuration, the data type for target system accesses is always byte. Otherwise, if you do Memory Display Byte, it will do byte reads, and if you do Memory Display Word, it will do word reads.

4 Press **<End>**, then **<Enter>** to save your changes and exit the configuration. Press **<Esc>** to discard your changes and exit the configuration.

The access mode affects interaction with target system memory. When you display or modify target system memory, the emulator has the HPC processor execute the monitor to read or write target memory locations. The access mode determines whether the emulator uses byte or word instructions for the memory accesses.

The **same** access mode option tracks the bus mode configuration item. (See "To set the bus configuration" in this chapter.) When this configuration item is enabled, the access mode is byte. If it is disabled, the access mode is the same as the display mode setting. (For example, the display mode is byte if you select Memory Display Byte or Memory Modify Byte.) This allows you to choose either byte or word accesses for memory modification if you have peripherals of different sizes in your target system. You need to examine your target system design and determine whether memory-mapped I/O peripherals are sensitive to byte or word writes, and choose the appropriate access mode.

To enable fast emulation of on-chip memory

1 Select Config Pod.

2 Use the arrow keys (if necessary) to move to the field next to the question "0 wait state execution?"

3 **Tab** to select **yes** if you want fast in-circuit emulation of on-chip memory in the range 8000..ffff hex. Select **no** to disable fast in-circuit emulation of on-chip memory.

4 Press **<End><Enter>** to save your changes.

Wait states are selected by program control via two bits in the PSW register. The READY line can be used to extend the bus cycle by adding wait states dynamically. Standard emulation memory in the HPC emulator requires one wait state.

Because of timing constraints, the HPC is seldom operated with external memory and zero wait states. However, applications which do not require off-chip memory can conveniently run with zero wait states. The zero wait state mode is provided for these applications.

When you select **yes**, 32 Kbytes of fast emulation memory is mapped to the range 8000..ffff hex. This memory allows you to run in-circuit at speeds up to 20 MHz with no wait states.

If the external clock speed is greater than 20 MHz, the emulator won't work correctly. It will not display a warning. If the processor is programmed for 0 wait states (via the PSW register), the emulator won't operate correctly with the internal clock unless the 0 wait configuration item is enabled.

Bank switching is not allowed in this mode. You can add only emulation memory map terms in the range 8000 through ffff hex. You cannot add target system RAM or ROM terms, since the HPC chip doesn't support external memory accesses when configured for on-chip ROM.

The processor and memory map are reset when you change this setting. This configuration item is always forced to **no** with the HP 64775J emulator,

because the HPC 16400E processor doesn't support on-chip ROM or zero-wait state operation.

To set the bus configuration

1 Select Config Pod.

2 Use the arrow keys to move to the field next to the question "8 or 16 bit external bus mode?"

3 **Tab** to select **8 bit** or **16 bit** bus mode as needed by your system and programs.

4 Press **<End>**, then **<Enter>** to save your changes and exit the configuration. Press **<Esc>** to discard your changes and exit the configuration.

You can set the HPC emulator for 8 bit or 16 bit out-of-circuit operation. If you set this configuration item to **8 bit**, the emulator pulls the /HBE line high on the rising edge of the RESET signal when out-of-circuit.

When in-circuit, the emulator reads the target system /HBE line to set the bus size. If the target system value differs from that set by the configuration, the emulator displays a status message. This condition isn't tested until the processor is released from reset. The target system setting of the /HBE line overrides the setting of this configuration item.

The HPC emulator probe has a separate control line that carries the emulator reset signal. You should connect this line to the target system reset in a wired-OR configuration. Then, when the emulator is reset, the target system is reset, and the HPC processor will be properly configured.

When you change the value of this configuration item, the processor is reset.

To enable ROMless mode

1 Select Config Pod.

2 Use the arrow keys to move to the field next to the question "Select Normal or ROMless mode?"

3 **Tab** to select **normal** to enable Normal mode for out-of-circuit operation. Select **ROMless** to enable ROMless mode for out-of-circuit operation.

4 Press **<End>**, then **<Enter>** to save your changes and exit the configuration. Press **<Esc>** to discard your changes and exit the configuration.

The HPC processor can operate in NORMAL or ROMless modes. In ROMless mode, the amount of memory defined as the on-chip ROM space is considered to be off-chip memory. (See the *HPC Terminal Interface User's Guide* for information on the **cf ocrom** configuration item, which sets the amount of on-chip ROM present.)

When you're operating in-circuit, the emulator checks the EXM input at the probe and sets the mode accordingly.

When you start a program run after an emulator reset, the emulator checks the EXM line at the target system interface and compares the value to this configuration item. If they differ, the emulator displays a status message. (See the previous configuration item for information on connecting emulator reset to target system reset.) The target system setting of the EXM line overrides this configuration item.

Since the HPC 16400E can't be operated in NORMAL mode, it always operates in ROMless mode. So, it doesn't have an EXM line. Then the only valid setting for the HP 64775J emulator is **ROMless**.

When you change the value of this configuration item, the processor is reset.

To enable/disable bank switching

1 Select Config Pod.

2 Use the arrow keys to move to the field next to the question "Enable NSC AN-497 bank switching?"

3 **Tab** to select **yes** if you want to use bank switching. Select **no** if you aren't using bank switching.

4 Press **<End>**, then **<Enter>** to exit the configuration and save your changes. Press **<Esc>** to exit the configuration and discard your changes.

When you select **yes** for this configuration item, the emulator allows memory bank switching as defined in National Semiconductor's application note AN-497. The standard model defined in AN-497 supports a memory range of 544 Kbytes. If you redefine the upper and/or lower bounds of the bank switch range (described in the configuration item "To define the bank switch address range"), the emulator will support a memory range up to a theoretical maximum of 962 Kbytes.

When you disable this configuration item, memory is limited to 64 Kbytes. You can't use bank switching if you've chosen the fast (zero wait state) emulation of on-chip RAM with the configuration. Then memory is limited to 32 Kbytes.

The other map term is set to **tram** if you select **yes**. It cannot be **eram** or **erom**. If you select **no**, the other map term defaults to **eram**.

The HPC emulator must have memory in the range ff00..ffff, so that it can load the vector table as necessary for program runs. When you're out of circuit, and have mapped other memory to anything but emulation memory, the emulator checks to see that some emulation memory (**erom** or **eram**) is in this range. Otherwise, it will display an error message.

When you change this configuration item, the processor and memory map are reset.

To define the bank switch address range

1 Select Config Pod.

2 Use the arrow keys to move to the field next to the question "Lower bound of bank range?"

3 Type in an address of the form XX00 to define the lower boundary of the bank switching address range.

4 Use the arrow keys to move to the field next to the question "Upper bound of bank range?"

5 Type in an address of the form XXff to define the upper boundary of the bank switching address range.

6 Press **<End>**, then **<Enter>** to exit the configuration and save your changes.

or

Press **<Esc>** to exit the configuration and discard your changes.

National Semiconductor's application note AN-497 suggests using address line A15 as the control between upper and lower memory banks. This gives 32 Kbytes in each bank (minus the on-chip register and RAM area).

By designing more complex bank switch decoding, you can increase or decrease the size of each bank. Your target system must include the proper decoding circuitry. When you map memory to emulation, the mapper uses the "Lower bound" and "Upper bound" settings to decode bank switches.

Your design should reserve enough memory in the common bank to contain all the interrupt service routines, common code, common data, and so on. Refer to AN-497 and the National Semiconductor Assembler/Linker manuals for more information.

The lower boundary defaults to the amount of on-chip RAM and registers plus one. It must be the lower end of a 256 byte boundary, and above the on-chip register and RAM area.

The upper boundary must be the high end of a 256 byte boundary, and must be below the value e000 hex.

When you change either of these configuration items, the processor and memory map are reset.

Examples

To allow 55 Kbytes of memory in each memory bank (minus the on-chip RAM and registers for the HPC 16003), define the lower boundary as 200 and the upper boundary as 0dfff.

To assign memory map terms

1 Select Config Map Modify.

2 Use the arrow keys to move to one of the "Address Range" fields next to a term.

3 Type in an address range of the form **<lower>..<upper>** which are the lower and upper boundaries of the term you want. The boundaries must be aligned on 256 byte boundaries.

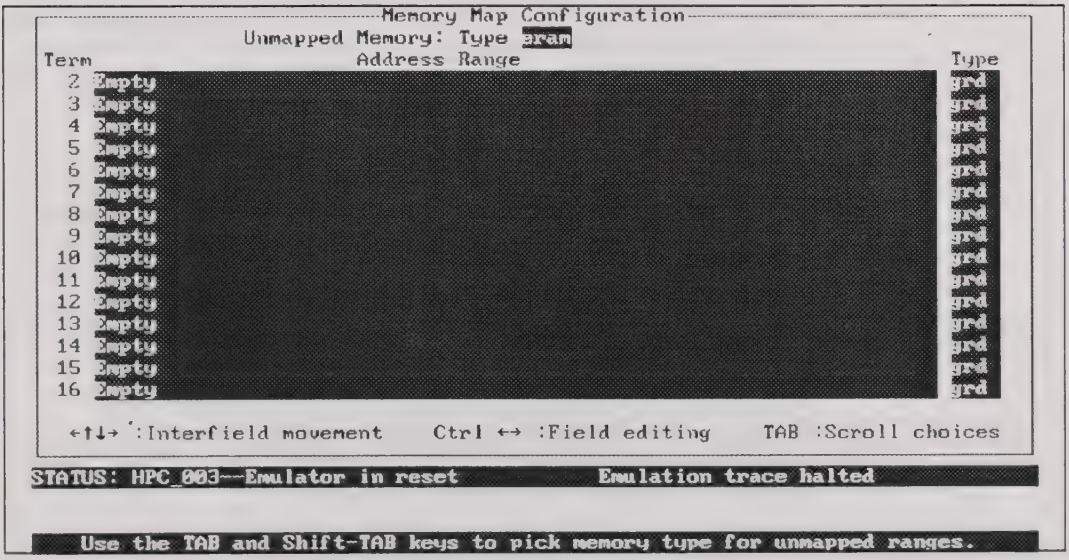
4 Press **<Enter>** to move to the "Memory Type" field for that term.

5 **Tab** to select a memory type as follows:

Type value	Memory Assigned
eram	Emulation RAM
erom	Emulation ROM
tram	Target System RAM
trom	Target System ROM
grd	Guarded memory

6 Repeat steps 2 through 5 for all the address ranges you want to assign to memory types.

7 Press **<End>**, then **<Enter>** to exit the map and save your changes. Press **<Esc>** to exit the map and discard your changes.



Config Map Modify Screen Form

You need to specify the location and type of various memory regions used by your programs and your target system. The emulator needs this information to:

- Translate addresses for emulation memory accesses.
- Pass through addresses for target memory accesses.
- Reserve emulation memory blocks.
- Set the type of memory blocks so that configuration items such as write to ROM break will operate correctly.
- Inhibit HPC writes to emulation ROM.

The HPC emulator has 16 map terms. One term is always reserved for the on-chip RAM definition. One term is used if you select a foreground monitor.

When you select the zero wait state emulation mode, you can only add emulation memory map terms in the range 8000 through ffff hex. You can't add target system RAM or ROM terms, because you're using the HPC chip in a mode that doesn't support external memory.

Your address specifications must begin and end on 256 byte boundaries. To specify an address beginning on a 256 byte boundary, enter an address ending in 00. To specify an address ending on a 256 byte boundary, enter an address ending in ff. If you enter a range that spans less than 256 bytes, the system will round the addresses as necessary to span a 256 byte block. (For example, the range 0..1f will be rounded to 0..ff).

If you want to add a term that overlaps address ranges with an existing term, you must either redefine or delete the existing term. If the term overlaps the term defined for the foreground monitor (which is not visible in the map), the term number displayed for overlap is the one known to the emulator, not to the PC Interface. To find the overlapping term, compare the address ranges of all terms to the base address of the foreground monitor (defined in a field next to the configuration item "Monitor type?").

Some things reset the memory mapper. These are: Config Map Reset, selecting 0 wait state operation, changing monitor type, or selecting bank switching. So, you should configure the emulator before you map memory. Otherwise, you must reenter the map commands.

Example

Suppose that you're using the emulator in-circuit, and that there is a 12-byte I/O port at c000 hex in your target system. Also, you want to emulate on-chip ROM, which starts at e000. Enter the following map commands:

```
0c000..0c0ff tram
0e000..0ffff erom
```

To assign the memory map default

1 Select Config Map Modify.

2 Use the arrow keys (if necessary) to move to the field next to "Unmapped Memory: Type."

3 **Tab** to select a memory type as follows:

Type value	Memory Assigned
eram	Emulation RAM
erom	Emulation ROM
tram	Target System RAM
trom	Target System ROM
grd	Guarded memory space

4 Press **<End>**, then **<Enter>** to save your changes and exit the memory map. Press **<Esc>** to discard your changes and exit the memory map.

The unmapped memory type term specifies all address ranges not otherwise covered by existing memory map terms. This can save you time in memory mapping.

Often you will want to be notified when the processor accesses a nonexistent memory location during a program run. You use the grd (guarded memory) term to do this. The emulator will break to monitor and display a message when a guarded memory access occurs.

If you're using the zero wait state emulation memory mode, the other map term can only be set to eram, erom or grd. If you're using bank switching, the other map term can only be set to tram, trom or grd.

Examples

Suppose you're working in-circuit with a target system that has some peripherals and ROM, but no RAM. Eventually, this system will have RAM in all other address ranges. You might assign map terms for the peripherals, and target ROM. Then set the unmapped memory type to **eram**.

To check the memory map

1 Select Config Map Modify to access the memory map.

2 When you're finished viewing the map, you can press either **<End> <Enter>** to exit (if you made changes and want to save them), or **<Esc>** (if you don't want to change the map).

To delete memory map terms

1 Select Config Map Modify to access the memory map.

2 Use the arrow keys to move to the "Address Range" field of the term you want to delete.

3 Use the spacebar to wipe out the address range definition for that term, then press **<Enter>**.

4 Repeat steps 2 and 3 for all terms you want to delete.

5 Press **<End>**, then **<Enter>** to exit the map and save your changes. Press **<Esc>** to exit the map and discard your changes.

If you have selected the foreground monitor, a memory map term is created for the monitor.

If you delete map terms that are between other terms, the other terms are renumbered when you save the map. This eliminates gaps in the term numbering sequence.

To reset the memory map

- Select **C**onfig **M**ap **R**eset.

This removes all memory map terms (except those for the foreground monitor and on-chip RAM) and resets the map. Make sure that you want to reset the map beforehand. If you want to remove one or two terms without resetting the map, see the previous section.

To select the emulation monitor

1 Select **C**onfig **E**mulator.

2 Use the arrow keys (if necessary) to move to the field next to the question "Monitor type?"

3 **Tab** to select **Background** if you want to use the background monitor.

or

Select **Foreground** if you want to use the foreground monitor.

4 If you selected the **Foreground** monitor in the previous step, a new field will appear. Press **<Enter>** to move to it. Type in the base address where the foreground monitor should be loaded.

5 Press **<End>**, then **<Enter>** to exit the configuration and save your changes. Press **<Esc>** to exit the configuration and discard your changes.

The emulation monitor is used to perform emulation functions such as display and modification of target system or on-chip memory.

The background monitor overlays processor address space and doesn't use any processor memory resources. However, interrupts (RESET, NMI and maskable) are disabled when the emulator is running in background. This condition may not be tolerable to some target system designs.

Interrupts can be enabled during foreground execution, which may make the emulator more transparent in some applications. The monitor retains the interrupt state at the time the break occurred.

To use the foreground monitor, you enter an address in the form XX00, where XX00 is the starting address of the monitor on a 256 byte boundary. The monitor uses exactly 400h bytes of memory. A memory map term is created to define emulation memory space for the monitor. You can't delete or modify this term with the Config **M**ap command. Instead, you must use the monitor configuration item to change the monitor's position (which will modify the term). Or, you can return to the background monitor, which will delete the term.

Also, you can't use any portion of the foreground monitor memory map term range for any purpose. Doing so will destroy the monitor. You'll need to reset the emulator with **P**rocessor **R**eset, which will also reload the foreground monitor.

When you select the foreground monitor, the memory map is reset since the map is affected. So, you should first select the monitor type, then map memory. The processor is always reset when you change monitor types.

The foreground monitor can only reside in emulation memory. It is downloaded to emulation memory when you exit the configuration and save it. If you also select 0 wait state operation, you must locate the foreground monitor in the memory range 8000..ffff hex.

If you have trouble with emulation monitor functions, you can reload the monitor. The background monitor is automatically reloaded whenever you enter the PC Interface. The foreground monitor is reloaded when you use the **P**rocessor **R**eset command. Both monitors are reinitialized when you use the **P**rocessor **R**eset command.

To define user monitor routines

1 Select Config Emulator.

2 Use the arrow keys to move to the field next to the question "Call user monitor reset routine?"

3 Tab to select **yes** if you want to have a user program routine executed whenever the monitor is entered from emulation reset. Then, press **<Enter>** to move to the next field, and type in the starting address of that user program routine.

or

Tab to select **no** to disable that user monitor routine.

4 Repeat steps 2 and 3 for the question "Call user monitor break routine?," which defines a routine that is executed once whenever the monitor is entered from a break.

5 Repeat steps 2 and 3 for the question "Call user monitor loop routine?," which defines a routine that will be executed once each time the monitor cycles through its main loop.

6 Repeat steps 2 and 3 for the question "Call user monitor exit routine?," which defines a routine that will be executed once when the processor exits the monitor and returns to the user program.

7 To exit the configuration and save your changes, press **<End>**, then **<Enter>**.

or

To exit the configuration and discard your changes, press **<Esc>**.

You can define four routines that are executed during monitor operations according to the above criteria. You should end each routine with the RET instruction. If your system uses bank switching, the routines must be located in the common memory range, not in one of the switched banks.

Defining or changing the address setting for any of these routines will reset the processor, since the monitor must be modified.

The "monitor loop" user routine definition specifies a routine to execute whenever the monitor passes through its main loop one time. This routine will not be executed whenever the emulator does a temporary break to monitor. For example, if you are running the user program and ask for display of an on-chip register, the emulator performs a temporary break. The monitor reads the register value, and the emulator returns immediately to the user program.

The subroutine code must obey several rules.

- Do *not* enable maskable interrupts.
- You must save and restore registers other than "Monitor Saved Registers" (defined below).
- You can use any of the "Monitor Saved Registers" (defined below), but their values will not persist after the subroutine RET.
- Should not contain a software breakpoint (JSRP X instruction, see "To define the software breakpoint vector").
- Must use the RET instruction to exit and return to the main monitor code.
- You *must* load the code as part of your program.

The "Monitor Saved Registers" (saved and used by the monitor) are A, B, K, X, SP, PC, bank code in Port B, C bit in PSW, GIE bit in ENIR, location zero (locz), and the frame pointer (at address 0be hex).

If your monitor subroutine fails to follow these rules, the emulator is likely to become "lost." You can reset the emulator (**P**rocessor **R**eset command) or initialize it (exit the PC Interface, cycle power on the emulator, and reenter the PC Interface). Before resuming emulation, find and correct the bad component of the monitor subroutine code.

Defining Break Conditions

The HPC emulator supports software and hardware breakpoints. These breakpoints allow you to break the processor to the monitor when a specified event occurs, such as reaching a particular address or completing a trace measurement. The configuration items and commands shown in this section allow you to control the conditions that will cause an emulator break.

To define the software breakpoint vector

1 Select **Config Pod**.

2 Use the arrow keys to move to the field next to the question "Software breakpoint vector?"

3 **Shift-Tab** to select **off** if you want to reserve all 16 JSRP vectors for the user application, or **tab** to select one of vectors 0-15 to use as the vector for software breakpoints.

4 Press **<End>**, then **<Enter>** to exit the configuration and save your changes.

or

Press **<Esc>** to exit the configuration and discard your changes.

The JSRP vectors are as follows:

Vector Number	Vector Address
0	FFD1-FFD0
1	FFD3-FFD2
2	FFD5-FFD4
3	FFD7-FFD6
4	FFD9-FFD8
5	FFDB-FFDA
6	FFDD-FFDC
7	FFDF-FFDE
8	FFE1-FFE0
9	FFE3-FFE2
10	FFE5-FFE4
11	FFE7-FFE6
12	FFE9-FFE8
13	FFEB-FFEA
14	FFED-FFEC
15	FFEF-FFEE

The HPC emulator uses the JSRP instruction to implement software breakpoints. When you define a breakpoint using the **B**reakpoints **A**dd or **B**reakpoints **S**et commands, the emulator saves the instruction at the specified address. Then it writes the JSRP instruction at that address.

If you allocate one of the JSRP vectors to the software breakpoint, the emulator will initialize the software breakpoint mechanism by writing an fff8 hex into the selected JSRP vector every time that you run code following an emulator reset. It then reads back the vector to verify the write. If the write failed, an error message is displayed, and the emulator retunrs to reset. So, if you have target system ROM in the range ff00..ffff hex, breakpoints won't work since the emulator can't set the vector. (The emulator can set the vector if the memory is mapped to emulation ROM (**erom**).)

When the JSRP instruction is encountered during user program execution, the processor gets the vector from the JSRP vector area. If the vector number is the one chosen for software breakpoints, the processor will fetch fff8 as the

next instruction address. When it then fetches from fff8, the emulator recognizes the breakpoint and jumps to the monitor.

The emulator replaces the JSRP instruction with the instruction saved earlier, and clears the breakpoint status.

The default setting is 0. If you change this configuration item, the processor is reset. You should remove any existing software breakpoints before changing the software breakpoint vector number. You can't change the value if there are enabled software breakpoints.

The JSRP instruction is used by National Semiconductor's bank-switching library routines. If you're using bank-switching, choose a JSRP vector that isn't assigned to a bank-switch routine. (See the National Semiconductor HPC Assembler/Linker Librarian manual.)

If your program has a JSRP instruction that corresponds to the selected vector:

- If software breakpoints are enabled, the emulator will break to monitor and display an "Undefined software breakpoints" error message.
- If software breakpoints are disabled, the emulator will break to monitor, but will not display an error message. This may be confusing, so if you aren't going to use the software breakpoints feature, you should select **off**.

To enable or disable write to ROM break

1 Select Config Pod.

2 Use the arrow keys to move to the field next to the question "Enable breaks on write to ROM?"

3 **Tab** to select **yes** if you want the emulator to break to monitor when a write to ROM is detected. Select **no** if you want to ignore these events.

4 Press **<End>**, then **<Enter>** to exit the configuration and save your changes. Press **<Esc>** to exit the configuration and discard your changes.

You can choose to enable or disable hardware breakpoints. For example, you might want to disable the write to ROM break temporarily. (Perhaps you need to see the next few bus cycles after the write in the trace list.)

The emulator always prevents a write to ROM from changing the contents of emulation memory. It cannot prevent writes to target memory, so if the memory at a particular location is actually RAM, the contents will be changed.

To enable an analyzer hardware break

1 Select Config Trigger to access the trigger hardware configuration.

2 Use the arrow keys to move to the "Emulator" field under the TRIG1 column. Tab until the arrows point toward "Emulator."

3 Use the arrow keys to move to the "Analyzer" field under the TRIG1 column. Tab until the arrows point away from "Analyzer."

4 Press **<End>**, then **<Enter>** to exit and save your changes. Press **<Esc>** to exit and discard your changes.

You can use the analyzer trigger to cause breaks on the analyzer trigger event. This can be useful if you want to stop program execution under certain conditions.

Software breakpoints are often used if you want to cause a break at a certain address. The advantage of analyzer breaks is that you can specify complex combinations of address, data and status values that must be found in a particular order before the break occurs. Also, you can use an analyzer break to cause a break on a data value.

Example

Enable the analyzer hardware break. Then, define an analyzer trigger as a particular data and status combination. Start the analyzer, and run the program. When the analyzer finds the selected data and status combination, it will force a break to the monitor.

To enable CMB interaction

1 Select Config Emulator.

2 Use the arrow keys to move to the field next to the question "Enable CMB interaction?"

3 **Tab** to select **yes** if you want this emulator to respond to synchronized runs and breaks via the CMB. Select **no** if you don't want this emulator to participate in coordinated measurements.

4 Press **<End>**, then **<Enter>** to save your changes and exit the configuration. Press **<Esc>** to discard your changes and exit the configuration.

The Coordinated Measurement Bus (CMB) allows you to use multiple emulators in a synchronized measurement. See the *PC Interface Reference* for more information.

To restrict to real-time runs

1 Select Config Emulator.

2 Use the arrow keys to move to the field next to the question "Enable real time mode?"

3 **Tab** to select **yes** if you want to restrict the emulator to real-time runs. Select **no** if you want to enable all emulator functions.

4 Press **<End>**, then **<Enter>** to exit and save your changes. Press **<Esc>** to exit and discard your changes.

The emulator uses the emulation monitor program to implement some features, such as register displays. When the HPC processor executes the

monitor, it is not executing your target system program. This may cause problems in target systems that need real-time program execution.

If you enable real time mode, the emulator will stop running user code only with the **Processor Reset**, **Processor Break**, **Processor Go**, and **Processor Step** commands. Commands such as **Register** that require a break to monitor are rejected if the processor is running user code. Also, the **Memory** commands will be rejected if the address argument specifies on-chip or target system memory when the processor is running user code.

This configuration item doesn't affect hardware breakpoints such as write to ROM, break on analyzer trigger or guarded memory access breaks. It also doesn't affect the emulator's response to software breakpoints.

When you select **no**, all commands are accepted.

Setting the Analyzer Configuration

The analyzer has many configuration items that allow you to specify clock speeds, trigger configuration, trace list format, and so on. These configuration items are described in the *Terminal Interface Reference* and the *Analyzer Terminal Interface User's Guide*.

The HPC emulator adds two configuration items that allow you to specify whether the emulation analyzer traces certain HPC processor-specific system activity. These are the hold acknowledge and user NMI trace configuration items.

To enable tracing of external DMA activity

1 Select Config Emulator.

2 Use the arrow keys to move to the field next to the question "Include hold acknowledge cycles in trace?"

3 **Tab** to select **yes** if you want hold acknowledge (HLDA) information in the analyzer trace list. Select **no** if you want to ignore HLDA activity for analyzer traces.

4 Press **<End>**, then **<Enter>** to exit the configuration and save your changes. Press **<Esc>** to exit the configuration and discard your changes.

Hold Acknowledge (HLDA) is the alternate function of port B line 7. When you enable this configuration item, the analyzer adds the designator "hlda" to each bus cycle where the processor HLDA line was asserted.

All on-chip DMA from the HPC 16400E processor appears in the trace list, no matter how this item is set.

You must use a standalone state analyzer to trace the address and data values associated with the DMA activity. (The HP 64700 Series external analyzer is not wide enough for full analysis of address and data activity.) The emulation

analyzer monitors the HPC internal bus and will not reflect the actual DMA activity when HLDA is asserted.

To enable tracing of user NMI monitor processing

1 Select Config Emulator.

2 Use the arrow keys to move to the field next to the question "Include NMI monitor overhead in trace?"

3 **Tab** to select **yes** to add monitor processing of user NMI requests to the trace list. **Tab** to select **no** to ignore monitor processing of user NMI requests.

4 Press **<End>**, then **<Enter>** to exit the configuration and save your changes. Press **<Esc>** to exit the configuration and discard your changes.

If you set the NMI mode configuration item to immediate or wait, the emulation monitor processes user NMI requests to save the HPC interrupt state. When NMI monitor overhead is not included in the trace list, this activity isn't seen by the emulation analyzer since it's part of background execution. But, you can add this processing to the trace list by selecting **yes** for this configuration item.

If you set this configuration item to **no**, the processing is included in the trace list only if you add all background monitor cycles to the trace list. You do this with the Analysis Format command. See chapter 3 for more information.

Using Command Files for Common Configurations

When you begin using the emulator, you may find it difficult to choose the correct configuration for successful operation. You can preset certain configuration items to values most appropriate for a particular situation. These procedures are described in this section.

To set the configuration for in-circuit emulation

1 Select System Command **\hp64700\demo\64775\incir.hpc <Enter>**.

2 If the target system clock is not functional, select the internal clock. (See "To select the processor clock source" in this chapter.)

When you use the emulator in-circuit, some configuration items need to be changed to get the emulator running easily. The procedure above sets specific configuration items to values that simplify initial target system plug-in for most situations. You may need to change other configuration items to match the needs of your target system. The command file:

- Disables zero wait state mode.
- Maps all other memory to target RAM.
- Selects external clock.
- Disables software breakpoints.

Also, the emulator has one map term defined for the on-chip RAM. Bank switching is disabled by the default configuration.

If you have changed other configuration items and are not sure that they are correct, you may want to reinitialize the emulator, then use the **incir** command file to set the above configuration items.

To set the configuration for in-circuit emulation and zero-wait state emulation memory

1 Select System Command **\hp64700\demo\64775\incir0w.hpc <Enter>**.

2 If the target system clock is not functional, or it exceeds 20 MHz, select the internal clock. (See "To select the processor clock source" in this chapter.)

The **incir0w** command file sets specific configuration items to values that are appropriate for most in-circuit emulation situations where zero wait state emulation of on-chip ROM is necessary. The command file:

- Enables zero wait state mode
- Maps all other memory to emulation RAM
- Selects external clock
- Selects JSRP 0 as the breakpoint instruction

Also, the emulator has one map term defined for the on-chip RAM. Bank switching is disabled by the default configuration.

If you have changed other configuration items and are not sure that they are correct, you may want to reinitialize the emulator, then use the **incir0w** command file to set the above configuration items.

To set the configuration for out-of-circuit emulation

1 Select System Command **\hp64700\demo\64775\outcir.hpc <Enter>**.

The **outcir** command file sets certain configuration items to values that are appropriate for most out-of-circuit emulation situations. The command file:

- Disables bank switching.

- Maps all other memory to emulation RAM.
- Selects internal clock; sets the EA bit=1, and selects one wait state.
- Selects JSRP 0 for software breakpoints and enables breakpoints.

Also, the emulator has one map term defined for the on-chip RAM. Zero wait state mode is disabled by the default configuration.

If you have changed other configuration items and are not sure that they are correct, you may want to reinitialize the emulator, then use the **outcir** command file to set the above configuration items.

To set the configuration for out-of-circuit emulation with bank switching

1 Select System Command **\hp64700\demo\64775\outcirbs.hpc <Enter>**.

The **outcirbs** command file sets certain configuration items to values that are appropriate for most out-of-circuit emulation situations that use bank switching as defined in National Semiconductor's application note AN-497.

The configuration items changed are as follows:

- Enables bank switching.
- Maps the subroutine and interrupt vectors to emulation RAM.
- Maps all other memory to target RAM.
- Selects internal clock; sets the EA bit=1, and selects one wait state.
- Selects JSRP 0 for software breakpoints and enables breakpoints.

Also, the emulator has one map term defined for the on-chip RAM. Zero wait state mode is disabled by the default configuration.

The bank switched memory range is defined by the "Lower bound of bank range?" and "Upper bound of bank range?" configuration items. The defaults are a lower bound of 200 hex (300 hex for the HPC16004 emulator) and an

upper bound of 7fff hex. You should set these to match the bank switched memory range in your target system design.

If you have changed other configuration items and are not sure that they are correct, you may want to reinitialize the emulator, then use the **outcirbs** command file to set the above configuration items.

5

Using the Absolute File Reader

How to prepare absolute files for use with the emulator

A file format reader comes with the PC Interface. The reader converts HP 64000 absolute files into two files that are usable with the PC Interface. This means that you can use the National Semiconductor language tools (version 2.1 or later) to create absolute files, then load those files into the emulator using the PC Interface.

The reader can operate from within the PC Interface or as a separate process. You may need to use it separately if there is not enough memory on your personal computer to operate the PC Interface and HP 64000 Reader simultaneously. You also can operate the reader as part of a "make file."

What the Reader Does

The reader uses the following files as input:

- The HP64000 reader uses the absolute file (<file.X>), the linker symbol file (<file.L>), and the assembler symbol files (<scr1.A>, <scr2.A>, ...).

The reader produces an absolute file and an ASCII symbol file.

The Absolute File

The file format reader creates an absolute file (<file>.hpa). This absolute file is a binary memory image, which is optimized for efficient downloading into the emulator.

The ASCII Symbol File

The ASCII symbol file (<file>.hps) produced by the reader contains global symbols, module names, local symbols, and, when using applicable development tools such as a "C" compiler, program line numbers. Local symbols evaluate to a fixed (static, not stack relative) address.

Note

You must use the required options for your specific language tools to include symbolic ("debug") information in the absolute and symbol files. Also, the reader uses path information contained in the .L file to locate the .A files. Version 2.1 of the National Semiconductor HPOMFHPC utility generates incorrect path names when the .L file and the source files are not located in the same directory.

The symbol file contains symbol and address information in the form:

```
 module_name1
 module_name2
 ...
 module_nameN
global_symbol1  00234
global_symbol2  00678
...
global_symbolM  00BCD
|module_name1|# 1234          00872
|module_name1|local_symbol1   00653
|module_name1|local_symbol2   00872
...
|module_nameN|local_symbolK   00986
```

Symbols are sorted alphabetically in the groups: module names, global symbols, and local symbols. The space preceding module names is required. Although formatted for readability here, a single tab separates symbol and address.

Line numbers will appear similar to a local symbol except that "local_symbolX" will be replaced by "#NNNNN" where NNNNN is a five digit decimal line number. The addresses associated with global and local symbols are specific to the processor for which the absolute files were generated.

You access line number symbols by entering the following on one line in the order shown:

module name
colon (:)
space
the word "line"
space (required)
the decimal line number

For example:

```
MAIN.C: line 23
```

Note

If your emulator can store symbols internally, symbols will appear in disassembly. When the line number symbol is displayed in the emulator, it appears in brackets. Therefore, the symbol "MODNAME: line 345" will be displayed as "MODNAME:[345]" in mnemonic memory and trace list displays.

The local symbols are scoped. This means that to access a variable named "count" in a function named "Foo" in a source file module named "MAIN.C," you would enter "MAIN.C:Foo.count". The following table summarizes how to enter symbol names.

How to Access Variables

Module Name	Function Name	Variable Name	You Enter
MAIN.C	Foo	count	MAIN.C:Foo.count
MAIN.C	bar	count	MAIN.C:bar.count
MAIN.C		line number 23	MAIN.C: line 23

Where the Reader Programs are Located

The reader programs are installed in the directory named \hp64700\bin by default. This directory must be in the environment variable PATH for the HP 64000 Reader and PC Interface to operate properly. This is usually defined in the "\autoexec.bat" file.

The following examples assume that you have "\hp64000\bin" included in your PATH variable. If not, you must supply the directory name when executing the Reader program.

Using a Reader from MS-DOS

The command name for the HP 64000 Reader is **RHP64000**. The command syntax is:

```
RHP64000 [-q] <filename>
```

-q This option specifies the "quiet" mode, which suppresses the display of messages.

<filename> This represents the name of the HP 64000 linker symbol file (file.L) for the absolute file to be loaded.

The following command will create the files "DEMO.HPA" and "DEMO.HPS":

```
C:\HP64700\DEMO\64775> RHP64000 DEMO.L
```

Using a Reader from the PC Interface

The HPC PC Interface has a file format option under the "Memory Load" command. After you select an option, the chosen reader will operate on the file you specify. The PC Interface will accept the absolute and symbol files produced by the Reader.

To use the Reader from the PC Interface:

1 Start the PC Interface.

2 Select "Memory Load." The memory load menu will appear.

3 Specify the file format as HP64000.

4 Select the default memory type to load (Target, Emulation or Both).

5 Specify a file in the appropriate format (such as <file>.L for HP64000 format). (DEMO.L, for example). (This is the linker symbol file.)

The file extension can be something other than those shown above, but cannot be ".HPA," ".HPS," or ".HPT."

Note <file>.HPT is a temporary file used by the reader to process the symbols.

Using the file that you specify (DEMO.L, for example), the PC Interface does the following:

- It checks to see if two files with the same base name and extensions .HPS and .HPA already exist (for example, DEMO.HPS and DEMO.HPA).

- If DEMO.HPS and DEMO.HPA don't exist, the PC Interface invokes the reader to create them. The new absolute file, DEMO.HPA, is then loaded into the emulator.
- If DEMO.HPS and DEMO.HPA already exist but the creation dates and times are earlier than that of DEMO.L, the PC Interface invokes the reader to recreate them. The new absolute file, DEMO.HPA, is then loaded into the emulator.
- If DEMO.HPS and DEMO.HPA already exist but the creation dates and times are later those for the file DEMO.L, the PC Interface does not invoke the reader. The current absolute file, DEMO.HPA, is then loaded into the emulator.

Note

Only the PC Interface does date/time checking. When you run the reader at the MS-DOS command line prompt, it will always update the absolute and symbol files.

When the reader operates on a file, a status message will be displayed showing which reader is in use. When the reader is finished, another message shows that the absolute file is being loaded.

The PC Interface executes the reader with the "**-q**" (quiet) option by default.

If the Reader Won't Run

If your program is very large, the PC Interface may run out of memory while attempting to create the database file. You must exit the PC Interface and execute the program at the MS-DOS command prompt.

Include RHP64000 in a Make File

You may want to incorporate the reader as the last step in your "make file," or as a step in your construction process. This to eliminate the possibility of having to exit the PC Interface due to space limitations described above. Also, the files necessary for emulation will then be updated automatically every time you modify your program.

6

Solving Problems

What to do when the emulator doesn't behave as expected

Sometime during your use of the emulator, you'll encounter a problem that isn't adequately explained by an error message or obvious target system symptoms. This chapter explains how to solve some of these more complex problems.

If the PC Interface won't start

- ☐ Make sure that you have the proper cable to connect your PC to the emulator. See the *HP 64700 Card Cage Installation/Service Guide.*

- ☐ Make sure that all switches on the HP 64700 Card Cage are set correctly. For a standard RS-232 connection to a PC, the COMM switches on the back of the Card Cage should be all set to 0, with switch 13 set to 1. If you're using an RS-422 connection, you will need different switch settings. See the *HP 64700 Card Cage Installation/Service Guide.*

- ☐ Make sure that the emulator is connected to power and that the power switch is on.

- ☐ Make sure that the 64700tab configuration file (usually installed in \hp64700\tables) matches the communications port number, communications parameters and emulator description for the port you are using and the HPC emulator. See the *PC Interface Reference* for more information on the 64700tab file.

If the emulator won't run

An incorrect emulator configuration will prevent the emulator from leaving reset.

Symptoms

The processor won't leave reset.

No commands seem to work, especially those affecting run control, such as Processor Go, Processor Step, and Processor Break.

The PC Interface provides four command files to help you solve this problem.

1 Exit the PC Interface using System Exit Unlocked.

2 Choose a command file that most closely matches your use of the emulator:

- If you're operating in-circuit with a functional external clock select: System Command **\hp64700\demo\64775\incir.hpc <Enter>**

  If the target system clock is not functional, select the internal clock. (See chapter 4, "Configuring the Emulator.")

- If the target system does not have external memory and needs to run with zero wait states, select: System Command **\hp64700\demo\64775\incir0w.hpc <Enter>**

  If the target system clock is not functional, or it exceeds 20 MHz, select the internal clock. (See chapter 4, "Configuring the Emulator.")

- If out-of-circuit without bank switching, type: System Command **\hp64700\demo\64775\outcir.hpc <Enter>**

- If out-of-circuit with bank switching, type: System Command **\hp64700\demo\64775\outcirbs.hpc <Enter>**

If one of these built-in macros solves your problem, you may want to study the macro definition to understand the source of the problem. Select System MS-DOS Command **type <filename> <Enter>** to view the command file named **<filename>**.

If the emulator won't run from reset

Some HP emulators will run from reset when out-of-circuit, because the monitor code can use the processor to generate a reset signal.

In the HPC emulator, run from reset means run from target reset. Because the HPC does not support a programmable reset signal, the Processor Go Reset command will wait forever when operated out-of-circuit.

The HPC run from reset is designed for in-circuit applications. The HPC run occurs on the trailing edge of a low true reset pulse, which must remain true for a minimum of 64 cycles of the CK2 clock.

When you use the internal clock, the Processor Go Reset command can be issued when the target system is powered off.

When you use the external clock, the Processor Go Reset command *must* be issued while the target system has power and is generating a valid clock. However, once you obtain the "Waiting for target reset" message—power to the target system can be cycled if this is what is required to generate a target reset.

If the emulator won't run from the monitor

☐ When you run a program from the monitor, the program starts at an incorrect entry point.

Don't modify the top of the stack using the Memory Modify command while the monitor is running, or during a temporary break to monitor while running user code (U> prompt). If you do this, the monitor will return to the wrong program counter value. The processor's PC is saved on the top of the stack when the monitor is entered. You can modify the PC with the Register Modify Single **pc <Enter> <value> <Enter>** command, or by specifying an adddress with the Processor Go Address or Processor Step Address command.

☐ You observe different behavior depending on whether you are running user code following an emulator reset or after a normal break to monitor.

The monitor has two states: monitor_after_reset and monitor.

Monitor_after_reset is always the background monitor. The reset value of the PC is determined by reading the contents of the reset vector at 0fffe:0ffff. The stack pointer always starts at 0002, and all other HPC reset conditions occur.

The monitor state is set by the "Monitor type?" configuration item (see chapter 4). The following registers are used by the monitor. Their values are saved when the break to monitor occurs, and are restored prior to running user code via either the **P**rocessor **G**o or **P**rocessor **S**tep commands.

> A, b, k, x, sp, pc, word at location zero, frame pointer word at 0be, C bit in the PSW register, GIE bit in ENIR register, and the four bank code bits in port B (0e3) if bank switching is enabled.

When you read or write any of the saved registers while running user code, or in the monitor, the saved copy of the register is automatically saved or modified.

☐ Your code in, or operand access from external memory (either emulation or target) does not execute properly following reset.

Remember that the EA bit in the HPC PSW register is automatically set to zero (0) whenever the HPC processor is reset. The EA bit needs to be "1" whenever you use external memory (HPC16003 and HPC16004 families only). External memory is the range from the top of on-chip RAM to the bottom of on-chip ROM.

☐ You see the following message while running in the monitor:

```
Unexpected software breakpoint
```

A nested break to the monitor has occurred. This has several different causes. For example, it might be caused by a software breakpoint in a monitor subroutine. Or, it may be a software breakpoint in an interrupt service routine while using the foreground monitor.

See the section "Handling Nested Monitor Breaks" later in this chapter.

If a register or target memory modify doesn't appear to work properly

☐ If you are running user code, each register is modified during a separate break to the monitor, and user code runs between each of these breaks. Also, target memory is subdivided into blocks. Each block is modified during a separate break to the monitor.

- You may need to halt execution of your code by using the Processor Break command to perform a permanent break to monitor to ensure that several registers or memory locations are modified simultaneously.
- If you are modifying external memory, ensure that the EA bit is set in the PSW register (HPC16003 and HPC16004 families only). External memory is the range from the top of on-chip RAM to the bottom of on-chip ROM.

If you can't set a software breakpoint or modify the reset vector

☐ Does your target system implement the high memory area as ROM? If so, the emulator can't write software breakpoints or modify the vector table.

Remap the memory range 0ff00..0ffff to **erom** and reload your software. Now you can modify memory from the emulator, and the emulator can modify the software breakpoint vector.

If you see unexplained states in the trace list

☐ Did you modify memory or perform some other monitor operation while the trace was in process? If so, any foreground operations made by the monitor (such as writing memory or accessing the stack) will appear in the trace list.

☐ Are you sure that the sequence, storage and trigger specifications are set up to exclude the states that you don't need?

If you are having problems running with 8-bit memory

☐ Make sure that the EA bit in the PSW register is set if you are using external memory (HPC16003 and HPC16004 families only).

☐ Make sure that the stack is located in on-chip RAM space (HPC16003 and HPC16004 families only).

☐ Make sure that your access mode is set to byte or same (see "To set the access mode" in chapter 4).

If you have problems disassembling a JID or JIDW instruction in memory

The two jump table instructions—JID and JIDW—have a variable length jump table. The emulator's inverse assembler does not know this length. It treats the first two bytes after 0cc hex (the JID opcode) as destination vectors, and the first two words after 0ec hex (JIDW opcode) as destination vectors. Any additional vectors are incorrectly disassembled as instructions.

If you need to step over and not through interrupt service routines

☐ If you use the background monitor to step through your code while interrupts are enabled, you will often find that you are stepping through your interrupt service routine.

If you configure the emulator to use the foreground monitor, you will not see the interrupt service routine being executed when you step through non-interrupt code.

If you can't trigger the analyzer

☐ Data reads from on-chip memory aren't visible to the analyzer. The analyzer knows that a read occurred because of the processor control lines, but can't see the data. Therefore, you can't trigger on a specific data value during a read from an on-chip memory location.

☐ Use **lowdata** instead of **data** to specify a byte data access.

If the emulator moves your code or data when you modify the emulator memory map

The emulator's mapping algorithm modifies the physical location of emulation memory addresses whenever you add or delete emulation memory from the emulator's memory map. You need to set-up the emulator memory map—including selection of the foreground monitor—before loading code and/or initializing data.

If a nested break into the monitor occurs

The HPC emulator provides a number of ways to create a nested break into the monitor:

- Hitting a software breakpoint (or just executing the software breakpoint JSRP op code) in an interrupt service routine when configured to use the foreground monitor.
- Hitting a software breakpoint while executing a user supplied monitor subroutine when configured to use either the background or foreground monitor.
- Executing the foreground monitor which has been modified (after loading) to contain the software breakpoint JSRP op code.
- Hitting a software breakpoint while executing the user NMI service routine when configured for either **immediate** or **wait** modes (see "To configure non-maskable interrupt sharing" in chapter 4).

In general, placing a software breakpoint in a user supplied monitor subroutine or a non-maskable interrupt service routine will "crash" the emulator, requiring recovery by exiting the PC Interface and cycling power to the emulator.

The emulator *is* designed to work with a nested monitor break caused by placing a software breakpoint in an interrupt service routine when configured to use the foreground monitor.

Symptoms

While in the foreground monitor, hitting a software breakpoint in an interrupt subroutine will result in the following message:

Unexpected software breakpoint

When you check the contents of the PC via the command:

```
Register Display Single pc <Enter>
```

you will find the PC pointing to the next byte following the software breakpoint JSRP.

To recover from this situation:

1 You must manually delete the software breakpoint via the command:

```
Breakpoint Clear Single <address> <Enter>
```

2 Then, decrement the PC by one byte:

```
Register Modify pc <Enter> <PC - 1> <Enter>
```

Remember that all registers now reflect the state of the HPC when the nested break occurred.

3 You can return to the original monitor state via:

```
Processor Go Pc
```

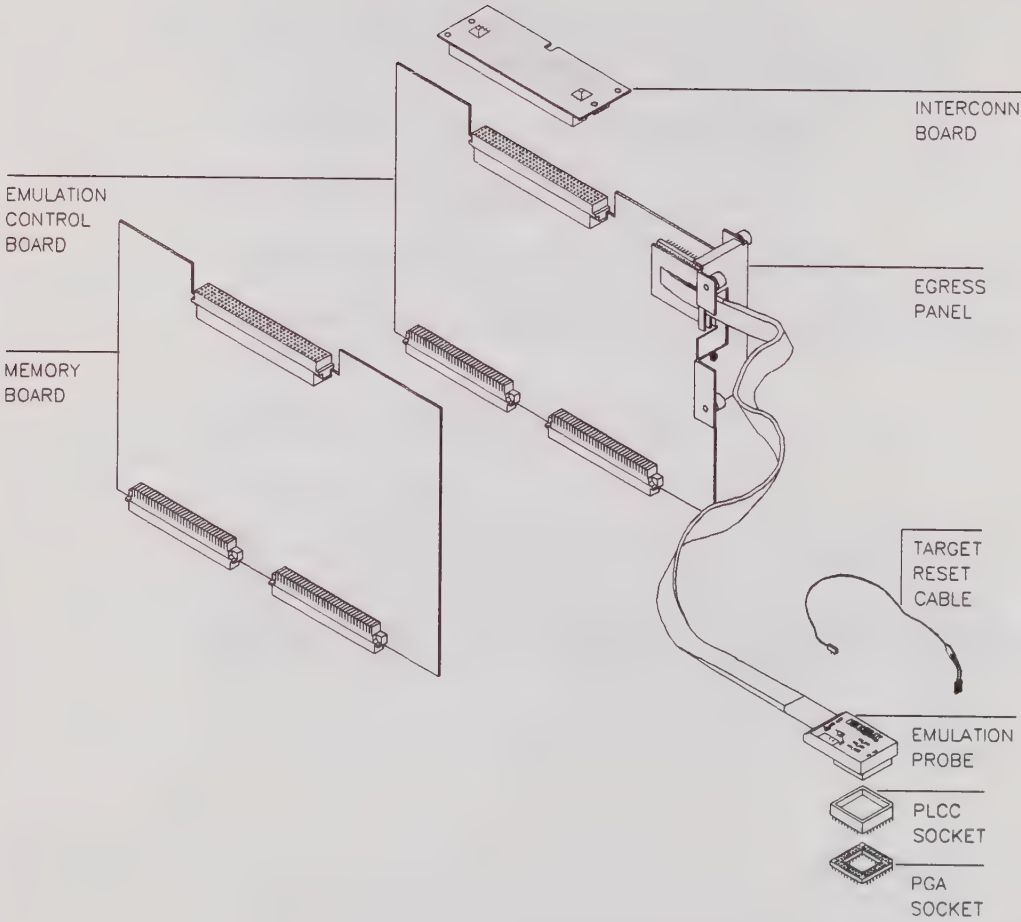
The nested break stayed in the background monitor until cleared via the Processor Go Pc command.

You must manually delete the software breakpoint and adjust the PC, because the emulator fails to handle a software break correctly while in the monitor.

7

Installation and Service

How to install and test the HPC emulator subsystem


INTERCONNECT BOARD
EMULATION CONTROL BOARD
EGRESS PANEL
MEMORY BOARD
TARGET RESET CABLE
EMULATION PROBE
PLCC SOCKET
PGA SOCKET

The HP 64775 emulator subsystem plugs into the HP 64700A Card Cage.

Equipment supplied

The HP 64775 package contains

- The HP 64775 emulator
- Transition Notice
- Terminal Interface User's Guide
- Terminal Interface Reference

This package with the PC Interface option contains

- PC Interface product software
- HP 64775 PC Interface user's guide

Support Services Information information is at the back of each binder.

Tools needed

- Flat blade screw driver
- Torx T-10 screw driver

Antistatic precautions

Computer cards contain electrical components that are easily damaged by small amounts of static electricity. To avoid damage to the emulator cards, follow these guidelines:

- If possible, work at a static-free workstation.
- Handle the parts only by the edges; do not touch components or traces.
- Use a grounding wrist strap that is connected to the computer's chassis.

Installation Overview

When you order a complete system (an HP 64775 emulator in an HP 64700A Card Cage, the HP 64775 emulator is already installed in the card cage. The installation procedure in this manual is provided for users that already have an HP 64700A Card Cage and want to install the HP 64775 emulator in the card cage, or would like to replace parts of the emulator.

Caution

When a customer already has a modular HP 64700A Card Cage and wants to remove the existing emulator and insert the HP 64775 emulator in its place, the HP 64700A generic firmware and analyzer firmware may NOT be compatible and will cause the emulator to NOT work. In this event, you must purchase a Flash EPROM board to update the firmware. Instructions for installing this board and programming it from a PC are provided in the *HP 64700A Card Cage Installation/Service* manual.

After hardware installation, the user should run a performance test to verify that the emulator is working properly. After you verify performance of the emulator, you are ready to update software, and use the emulator.

Use this section to accomplish hardware removal and installation procedures, running performance verification, and ordering parts. Refer to the *HP 64700A Installation/Service* manual for information on system configurations, installing product software, software updates, and ordering parts related to the card cage.

Note

Make sure that power to the HP 64700A Card Cage is off before removing or installing any hardware.

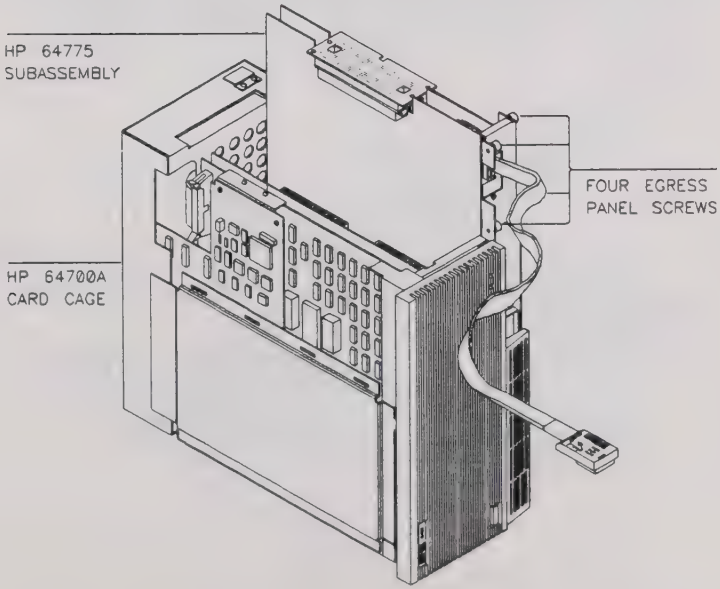
Caution

Possible damage to the emulator! When you use the emulator with a target system, always turn on power to the emulator before turning on power to the target system. Also, turn off power to the target system before turning off power to the emulator. Otherwise, the emulator might be damaged.

To install the emulator into the HP 64700A Card Cage

- Turn Power off.
- Remove the HP 64700A Card Cage top cover.
- Remove the HP 64700A Card Cage side panel.
- Install the HP 64775 emulator subassemblies and secure in place with the four Egress Panel screws.
- Replace the HP 64700A Card Cage side panel and top cover.

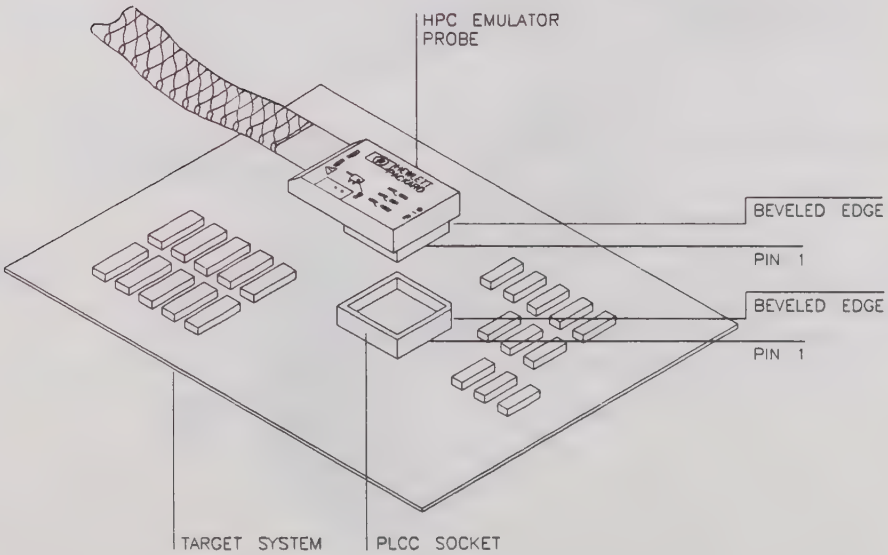
Instructions for removing and installing cards into and out of the HP 64700A Card Cage are provided in the HP 64700A Installation/Service manual and on the HP 64700A Card Cage label on the bottom of the HP 64700A Card Cage.



To install the emulator probe into a PLCC socket

- Turn off power to the target system, then turn off power to the emulator.
- Remove the PLCC to PGA adaptor supplied with the emulator probe.
- Locate pin 1 and the beveled edge on the emulator probe and the target PLCC socket.
- Insert the emulator probe into the target PLCC socket.
- Connect the target reset user cable from the emulator probe to the target system.
- Turn on power to the emulator, then turn on power to the target system.

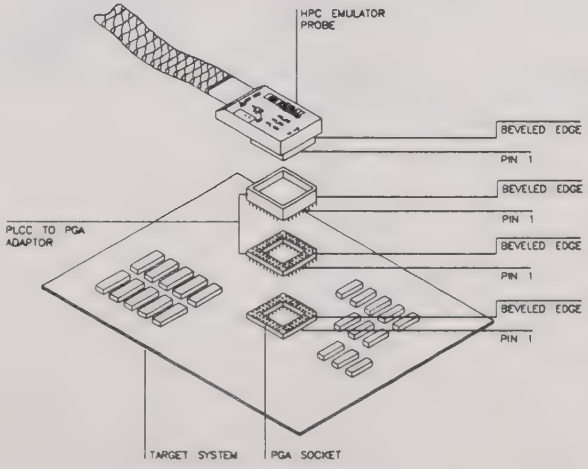
Make sure that pin 1 of the PLCC socket and pin 1 of the emulator probe are aligned. Also, the beveled edge on the emulator probe must be aligned with the beveled edge of the PLCC socket.



To install the emulator probe into a PGA socket

- Turn off power to the target system, then turn off power to the emulator.
- Locate pin 1 and the beveled edge on the emulator probe and the supplied PLCC socket.
- Attach the supplied PLCC to PGA adaptor to the emulator probe PLCC socket making sure that the beveled edges are aligned.
- Insert the emulator probe into the PGA socket of your target system making sure the beveled edges of the emulator cable are aligned with the beveled edge of the target PGA socket.
- Turn on power to the emulator, then turn on power to the target system.

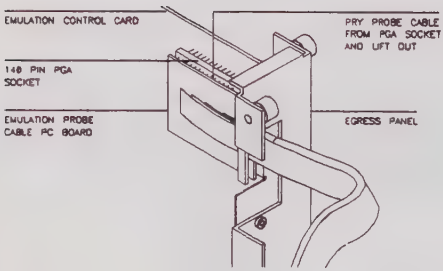
  When inserting the emulator cable into a PGA socket, make sure that there are no bent pins. It is possible to insert the emulator probe into the PGA socket only using the PLCC socket; however, the PLCC socket has square pins and may be difficult to insert into your target's PGA socket.

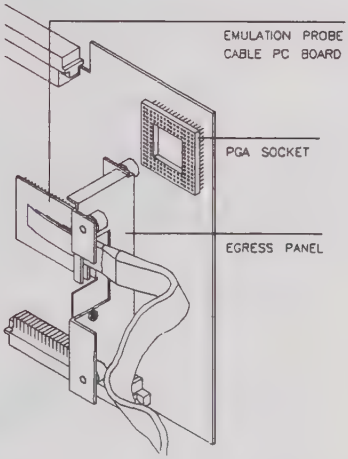


To remove the probe cable from the emulation control card

- Insert a flatblade screwdriver between the emulation probe cable PC board and the PGA socket.
- Pry the emulation probe cable from the PGA socket and lift off.

The egress panel should remain attached to the emulation probe cable.


EMULATION CONTROL CARD
140 PIN PGA SOCKET
EMULATION PROBE CABLE PC BOARD
PRY PROBE CABLE FROM PGA SOCKET AND LIFT OUT
EGRESS PANEL



To verify the performance of the emulator

- Establish communication with the emulator from your host or ASCII terminal.
- Turn on power to the HP 64700A Card Cage. You should see a "R>" prompt.
- Enter: **pv n** <return>

  where **"n"** is the number of cycles for running performance verification (pv).

  The performance verification (pv) test verifies that internal emulation circuitry is working properly. If your emulator is working in out-of-circuit (not connected to your target system), but does not work when connected to your target, the problem could be in the probe cable because pv does not test the cable.

  There are different hardware system configurations for the HP 64700 Series system. For information on hardware configurations, refer to the HP 64700A Installation/Service manual. The interface you are using depends on the host used. The **pv** command is a Terminal Interface command. If you are using the Terminal Interface, simply enter the command

  pv 1 <return>

  If you are using the PC Interface, you can still run performance verification by entering:

  System Terminal tcf -e <return> pv 1 <return>

Example

Invoke the Selftest routines through the command:

```
R>pv n
```

(n=1,2,3,.... specifies the number of times to run the set of performance verification tests.)

Note

It is sometimes necessary to enter the command **tcf -e** before running pv.

A message similar to the following should appear:

```
Testing:  HP64775G National Semiconductor HPC Emulator
    PASSED:
    Number of tests:  1           Number of failures:  0
  Testing:  HP 64740  Emulation Analyzer
    PASSED:
    Number of tests:  1           Number of failures:  0

                  Copyright (c) Hewlett-Packard Co. 1987
All Rights Reserved.  Reproduction, adaptation, or translation
without prior
written permission is prohibited, except as allowed under copyright
laws.

  HP64700 Series Emulation System
  Version:   A.03.01 13Mar91

  HP64775G National Semiconductor HPC Emulator
  HP64740 Emulation Analyzer
```

If you have an emulation failure, you can replace the entire assembly through your local Hewlett-Packard representative, and through the Support Materials Organization (SMO).

For component parts, refer to the parts list at the end of this chapter.

What is pv doing to the Emulator?

Self test procedures provide a thorough check of the functionality of all of the products installed in the HP 64700A Card Cage. The Test Suite for the HP 64775 Emulator consists of the following modules.

```
Test #          Function Test Name
   0       Host Tests
   1       Reset Release Tests
   2       Transition Tests
   3       Dual Port Access Tests
   4       Emulation Memory Banks Test
   5       Analysis Interface
```

Troubleshooting

The test results for all of these modules are indicated by a simple PASS/FAIL indication. The PASS message gives a high level of confidence that all major functions and signals are operating.

A FAIL message on the other hand indicates that one or more of the tested functions are NOT working. In this event an HP field representative can either swap assemblies to isolate the failure to an individual board, or replace all the major assemblies shown in the replaceable parts list.

To insure software compatibility

There are various sets of firmware resident in the assemblies contained in the HP 64700A Card Cage. The versions of these code modules could change occasionally and it is important to insure that all the versions are compatible between the group of products that you have installed.

There are four assemblies that have firmware in the HP 64700A Card Cage. These assemblies are the:

- Host Controller card
- Emulator cards
- Analyzer card

If you purchased a complete Emulation/Analysis System from HP, you can be assured that all the products contained in the HP 64700A Card Cage contain compatible firmware at the time of sale. Software compatibility problems can occur when you replace the host controller card, emulator card, or analyzer card from one HP 64700A Card Cage to another, or from a recently purchased subassembly.

For example, you might purchase only the emulator subassembly and replace the original emulator subassembly with the one you just purchased. In this case, the host controller may contain a version of firmware that is older than required to operate the new emulator; hence, compatibility problems are caused by a newer emulator needing newer generics. All emulators will work with the latest generics.

This emulator has Flash EPROMs that can be updated with current versions of firmware. Other products (assemblies) that do not use the Flash EPROM technology can also be updated with the latest firmware by adding a Flash EPROM card that is available from Support Materials Organization (SMO). A Flash EPROM card can be inserted in an available slot in the card cage to override old versions of firmware in products with conventional EPROMS. The host controller in the card cage is already programmed to look for a Flash EPROM card if one is inserted into the card cage.

The latest version of firmware for the host controller card, and analyzer card along with a program called *progflash* is sent to you on separate 5 1/4 inch diskettes and is identified by the word firmware printed on the label for PCs. Current versions of emulation firmware are part of PC Interface software.

When you load all your new versions of software onto your host computer, you are now ready to load the new version of firmware from your host computer to the assemblies that are in the HP 64700A Card Cage.

To load the new firmware, use the *progflash* command. The progflash command displays a list of card cages, and subassemblies in each card cage on your system. From these lists, you can select which product to update. For information on using the *progflash* command, and updating your HP 64700 Series firmware, refer to the HP 64700 Series Card Cage Installation/Service manual.

Parts List

Main Assembly	Component Parts	New	Exchange
interconnect board		64772-66505	
probe cable - NS16XX		64775-61601	
NS-HPC16XX3-emulator card		64775-66502	64775-69502
NS-HPC16XX4-emulator card		64775-66503	64775-69503
NS-HPC400E-emulator card		64775-66506	64775-69506
	PGA socket	1200-1253	
	PLCC socket	1200-1274	
	egress panel	64700-00203	
	emulator target reset coaxial cable asy.	64762-61602	
NS-HPC16XXX-memory card		64775-66508	64775-69508

What is an Exchange Part?

Exchange parts are shown on the parts list. Defective parts can be returned to HP for repair in exchange for a rebuilt part through the Support Materials Organization (SMO).

Emulator Card

There are three emulator cards that can be returned for repair and replaced on the exchange program. To replace the emulator control card on the exchange program, you must remove certain parts, and return only that part considered an exchange part. When returning the emulator control card, you must remove the:

- emulator probe cable
- interconnect board

Memory Card

Before returning the memory card on the exchange program, you must remove the interconnect board.

8

Emulator Messages

The information displayed by the emulator to tell you that something's wrong

This chapter has information on messages that are specific to the HPC emulator. Each message is listed with its error number and message text. The cause and remedy for each message follows that message.

0xh is outside the legal bank code range

The legal set of bank code values is 0..0f. The emulator rejects any bank code value greater than 15 decimal. Reenter the address specification with a bank code value in the range 0..0f.

Bank switching and zero wait state are mutually exclusive

You set "0 wait state execution?" to **yes** and "Enable NSC AN-497 bank switching" to **yes** (see chapter 4). You can't use the fast emulation memory mode with bank switching. The fast emulation memory mode is primarily designed for accurate emulation of on-chip ROM, without external memory. Bank switching is for systems that have large external memory requirements. Determine which configuration is appropriate for your target system and enable only one.

Bank switching enabled; invalid bshigh value

The bank switch high address value ("Upper boundary of bank switch range?") must be below 0e000 hex. Use this configuration item to specify a legal upper boundary value.

Bank switching enabled; invalid bslow value

The bank switch low address value ("Lower boundary of bank switch range?") must be above the on-chip RAM and register memory range. Use this configuration item to specify a legal lower boundary value.

Bank switching is NOT enabled, "bc" reg is not defined

You asked to read or write the bank code register but bank switching is not enabled. You can configure the emulator to enable bank switching by setting the configuration item "Enable NSC AN-497 bank switching?" to **yes**.

bp functionality removed via "cf swbpvtr=none"

You can't add, set, clear or remove software breakpoints if "Software breakpoint vector?" is set to **none**. You need to select a software breakpoint

vector using this configuration item before you can successfully use the Breakpoint command options.

Bus cycle error when loading mapper

The emulation system controller had an error while loading the emulation memory mapper. Try reinitializing the emulator (exit the PC Interface, cycle power, and reenter). Reenter the Config Map Modify command, and reenter the memory map terms. If the problem is repeated, contact your local HP Sales and Service office for assistance.

Disable all breakpoints before changing the bp JSRP

You used the "Software breakpoint vector?" configuration item to change the software breakpoint vector, but there were existing software breakpoints in memory. If you changed the vector without disabling these breakpoints, the emulator would not handle them correctly. Disable all breakpoints with the Breakpoints Remove All command, then change the software breakpoint vector.

Emulator failed to initialize correctly

The HP 64700 host controller could not correctly initialize the emulator. Try cycling power to see if initialization will work correctly. Or, use the **pv** command to verify emulator operation. (See chapter 7, "Installation and Service.")

FGD monitor must be located outside bank switch range

You selected **Foreground** in the "Monitor type?" configuration item, and set the **<address>** value between the lower and upper boundaries of the bank switch range with bank switching enabled. The foreground monitor must be located in common memory, outside the bank switch range. Reenter the **<address>** value for the monitor with an address outside the range "Lower boundary of bank switch range?" to "Upper boundary of bank switch range?"

Foreground monitor must be located above 07fff when "cf 0_wait=en"

You selected the fast emulation memory mode by answering **yes** to the "0 wait state execution?" configuration item, but have put the foreground monitor below 8000 hex with the **<address>** specifier. When you use fast emulation memory mode, memory is available only from 8000 through ffff hex. Either

disable the fast emulation memory mode, or move the foreground monitor above 8000 hex.

Foreground monitor must be located above on-chip RAM range

You selected **Foreground** for the "Monitor type?" configuration item, but you selected an **<address>** value in the on-chip register and RAM range. You must place the monitor in emulation memory. Enter an **<address>** value that can be mapped to emulation memory.

Host bus error detected

The emulation system controller had a bus error during interaction with the HPC emulator. Try reinitializing the emulator (exit the PC Interface, cycle power to the emulator, and reenter the PC Interface). Repeat the sequence of operations that leads to the error. If the problem is repeated, contact your local HP sales and service office for assistance.

HPC vector space at 0ffd0..0ffff mapped to GRDed

You tried to run the HPC processor, but the 256 byte address range 0ff00..0ffff is mapped to guarded memory (**grd**). The processor can not run in this situation. Remap the address range from 0ff00..0ffff to emulation or target system memory.

Invalid monbrk address

You selected **yes** for the "Call user monitor break routine?" configuration item, and specified an invalid address expression or one that falls in an illegal range:

- (**cf ocram** value + 1)..7fff when "0 wait state execution?" is set to **yes**.
- Between the addresses specified for "Lower boundary of bank range?" and "Upper boundary of bank range?" when "Enable NSC AN-497 bank switching?" is set to **yes**.
- (foreground monitor base address)..(foreground monitor base address + 3ff) when the **Foreground** monitor is selected for "Monitor type?" and is set to **<address>** (where **<address>**=foreground monitor base address)
- HPC register range (0c0..1bf)

Enter a valid address expression in a legal memory range (outside the ranges given above).

Invalid monexit address

You selected **yes** for the "Call user monitor exit routine?" configuration item, and specified an invalid address expression or one that falls in an illegal range:

- (**cf ocram** value + 1)..7fff when "0 wait state execution?" is set to **yes**.
- Between the addresses specified for "Lower boundary of bank range?" and "Upper boundary of bank range?" when "Enable NSC AN-497 bank switching?" is set to **yes**.
- (foreground monitor base address)..(foreground monitor base address + 3ff) when the **Foreground** monitor is selected for "Monitor type?" and is set to **<address>** (where **<address>** = foreground monitor base address)
- HPC register range (0c0..1bf)

Enter a valid address expression in a legal memory range (outside the ranges given above).

Invalid monloop address

You selected **yes** for the "Call user monitor loop routine?" configuration item, and specified an invalid address expression or one that falls in an illegal range:

- (**cf ocram** value + 1)..7fff when "0 wait state execution?" is set to **yes**.
- Between the addresses specified for "Lower boundary of bank range?" and "Upper boundary of bank range?" when "Enable NSC AN-497 bank switching?" is set to **yes**.
- (foreground monitor base address)..(foreground monitor base address + 3ff) when the **Foreground** monitor is selected for "Monitor type?" and is set to **<address>** (where **<address>** = foreground monitor base address)
- HPC register range (0c0..1bf)

Enter a valid address expression in a legal memory range (outside the ranges given above).

Invalid monrst address

You selected **yes** for the "Call user monitor reset routine?" configuration item, and specified an invalid address expression or one that falls in an illegal range:

- (**cf ocram** value + 1)..7fff when "0 wait state execution?" is set to **yes**.
- Between the addresses specified for "Lower boundary of bank range?" and "Upper boundary of bank range?" when "Enable NSC AN-497 bank switching?" is set to **yes**.

- (foreground monitor base address)..(foreground monitor base address + 3ff) when the **Foreground** monitor is selected for "Monitor type?" and is set to **<address>** (where **<address>**=foreground monitor base address)
- HPC register range (0c0..1bf)

Enter a valid address expression in a legal memory range (outside the ranges given above).

Modify bank code register to run code in switched memory

You entered a run command using a bank code as part of the run address (for example: Processor Go Address **X:XXXX <Enter>**). Instead, you must define the bank code using the command Register Modify **bc <Enter> <bank code> <Enter>**, then use the command Processor Go Address **xxxx <Enter>** where **xxxx** is the offset in the selected bank.

Nested monitor break, run from reset not supported

The emulator is currently operating in a nested break to the monitor. This nested break condition must be closed before the emulator will respond to a Processor Go Reset command. See chapter 5 for more information on nested break conditions.

No target Vcc, but 0ffd0..0ffff mapped to TRAM

No target Vcc, but 0ffd0..0ffff mapped to TROM

The HPC cannot run from reset unless the reset vector contains a useful value. You have mapped 0ff00..0ffff to **tram** or **trom** but the emulator is unable to detect target Vcc following a run or step command.

Other map term changed to tram since bank switching enabled

You selected **yes** for the configuration item "Enable NSC AN-497 bank switching?" to enable memory bank switching. The emulator simply informs you that the memory map default was automatically changed to target system RAM (**tram**).

Other map term changed to eram for Zero wait state emulation

You selected **yes** for "0 wait state execution?" to enable fast emulation of on-chip memory. The emulator simply informs you that the memory map default was automatically changed to emulation RAM.

Permanent break timeout while in FGD monitor

The emulator believes that the HPC is running the foreground monitor, but the emulation system controller did not obtain a timely handshake response from the foreground monitor.

Permanent break timeout while in BGD monitor

The emulator believes that the HPC is running the background monitor, but the emulation system controller did not obtain a timely handshake response from the background monitor.

Software breakpoint access error

The emulator is unable to access a software breakpoint. You must reinitialize the emulator. Exit the PC Interface, cycle power on the emulator, and reenter the PC Interface.

Software breakpoint can't be placed in foreground monitor

You used the **B**reakpoints **A**dd or **B**reakpoints **S**et commands with an address that falls in the foreground monitor address space. You can't place breakpoints here. Also, you should not place breakpoints in the user monitor subroutines, or your NMI service routine. Reenter the **B**reakpoints command with an address value corresponding to a legal memory range.

Software breakpoint can't be placed in HPC vector table

You used the **B**reakpoints **A**dd or **B**reakpoints **S**et commands with an address value in the range ffd0 thorugh ffff hex. This address range is the HPC processor's subroutine and interrupt vector table. You can't place breakpoints here. Reenter the **B**reakpoints command with an address value corresponding to a legal memory range.

Software breakpoint can't be placed in locz (0..1)

You used the **B**reakpoints **A**dd or **B**reakpoints **S**et commands with an address of 0 or 1. You can't place breakpoints here, because when the HPC is released from reset, this word is initialized to all zeros. Reenter the **B**reakpoints command with an address value corresponding to a legal memory range.

Software breakpoint can't be placed in on-chip register space

You used the **B**reakpoints **A**dd or **B**reakpoints **S**et commands with an address that falls in the on-chip register space (0bc..01bf). You can't place breakpoints here. Reenter the **B**reakpoints command with an address value corresponding to a legal memory range.

Software breakpoint while in foreground monitor

A software breakpoint was encountered while the emulator was running the foreground monitor. The emulator is placed in the background monitor to prevent repeated foreground monitor entries. See chapter 5 for more information about nested breaks to the monitor.

Software breakpoint while in user's NMI service routine

A software breakpoint was encountered while the emulator was running a user NMI service routine. The emulator is placed in the background monitor. See chapter 5 for more information about nested breaks to the monitor.

Software breakpoint while in user's monitor subroutine

A software breakpoint was encountered while the emulator was running a user monitor subroutine (**monloop**, **monbrk**, **monrst**, or **monexit**). The emulator is placed in the background monitor. See chapter 6 for more information about nested breaks to the monitor.

Target is holding RESET low

The emulator is unable to run or step because the target system is holding the processor in reset. Release the target system reset, then repeat the run or step command.

Target signals 8 bit memory, but "cf 8bit=dis"

You're using the emulator in-circuit. The HBE line in the target system is high when the processor is released from reset, indicating that the target system is running a 8-bit bus. However, the emulator configuration item "8 or 16 bit external bus mode?" is set to **16 bit**. Check your target system requirements, and change the configuration to **8 bit** if appropriate.

Target signals 16 bit memory, but "cf 8bit=en"

You're using the emulator in-circuit. The HBE line in the target system is low when the processor is released from reset, indicating that the target system is running a 16-bit bus. However, the emulator configuration item "8 or 16 bit external bus mode?" is set to **8 bit**. Check your target system requirements, and change the setting to **16 bit** if appropriate.

Target signals EXM bit TRUE, but "cf exm=dis"

You're using the emulator in-circuit, and the emulator sees the EXM input from the target system is true when the processor is released from reset. However, the emulator configuration item "Select Normal or ROMless mode?" is set to **normal**. Check your target system configuration, and change the setting to **ROMless** if appropriate.

Target signals EXM bit FALSE, but "cf exm=en"

You're using the emulator in-circuit, and the emulator sees the EXM input from the target system is false when the processor is released from reset. However, the emulator configuration item "Select Normal or ROMless mode?" is set to **ROMless**. Check your target system configuration, and change the setting to **normal** if appropriate.

The "sp" register can NOT contain an odd value

The HPC stack pointer operates on addresses only as 16 bit words. The HPC does not support word accesses on odd byte boundaries. If the stack pointer value is odd, operation of the HPC processor is unpredictable.

Unable to detect Vcc from target

You have configured the emulator to use an external clock by selecting **no** for the "Select internal clock source?" configuration item, but the emulator is unable to detect target system Vcc when instructed to run or step user code. Make sure that the emulator probe is properly installed (see chapter 2). Also make sure that target system power is on.

Unable to initialize swbpvtr at 0ffxx

The emulator attempted to initialize the subroutine vector chosen for breakpoints to 0fff8, but was not able to read back 0fff8 from the same location. If memory at the subroutine vector location is mapped to target

system ROM, the memory range 0ff00..0ffff must be mapped to emulation memory and properly initialized before software breakpoints can be used. Or, you can disable software breakpoints by selecting **off** for the configuration item "Software breakpoint vector?"

Unable to properly control emulator

The emulator failed to perform basic control functions correctly. Run performance verification software using the **pv** command to test emulator operation. (See chapter 7, "Installation and Service.") Also, try cycling card cage power to see if the problem is cleared.

Use "cf mon" to change foreground monitor

While in the Terminal Interface (System Terminal), you entered a **map <range> eram fg_mon** command, or you attempted to use the **map -d** command to delete the foreground monitor term (marked with attribute **fg_mon**). You can't use the **fg_mon** attribute in the **map** command. This is reserved for the emulator. If you want to change the location of the foreground monitor, change the **<address>** value next to the "Monitor type?" configuration item in the PC Interface. (See chapter 4 for details.)

Use "cf ocram" to modify on-chip RAM definition

While in the Terminal Interface (System Terminal), you entered a **map <range> tram on-chip** command, or you attempted to use the **map -d** command to delete the on-chip ram term (marked with attribute **on-chip**). You can't use the **on-chip** attribute in the **map** command. This is reserved for the emulator. If you're using an HPC family processor that isn't directly supported by the emulator, and you need to change the size of the on-chip RAM area, use the **cf ocram** command in the Terminal Interface. (To access the Terminal Interface, select System Terminal.)

Word access of target memory is not permitted

You selected **8 bit** for the "8 or 16 bit external bus mode?" configuration item. This configures the emulator for a target memory bus 8 bits wide. But you have requested 16 bit target memory access by selecting **word** for the "Target memory data access width?" configuration item. Study your target system design. If the target system memory data bus is only 8 bits wide, select **byte** or **same** for the "Target memory data access width?" configuration item.

You modified "psw" reg at 0c0 to 0xx

You used the Memory Modify command with an address of **0c0** to modify the psw (processor status word). This message is displayed as a status message only, since this action may not have been what you intended.

You modified "sp" reg at 0c4 to 0xxxx

You used the Memory Modify command with an address of **0c4** to modify the sp (stack pointer). This message is displayed as a status message only, since this action may not have been what you intended.

You modified "pc" reg at 0c6 to 0xxxx

You used the Memory Modify command with an address of **0c6** to modify the pc (program counter). This message is displayed as a status message only, since this action may not have been what you intended.

You modified low byte of "sp" register at 0c4 to 0xx

You used the Memory Modify command with an address of **0c4** to modify the lower byte of the sp (stack pointer) register. This message is displayed as a status message only, since this action may not have been what you intended.

You modified low byte of "pc" register at 0c6 to 0xx

You used the Memory Modify command with an address of **0c6** to modify the lower byte of the pc (program counter) register. This message is displayed as a status message only, since this action may not have been what you intended.

You modified high byte of "pc" register at 0c7 to 0xx

You used the Memory Modify command with an address of **0c7** to modify the upper byte of the pc (program counter) register. This message is displayed as a status message only, since this action may not have been what you intended.

You modified high byte of "sp" register at 0c5 to 0xx

You used the Memory Modify command with an address of **0c5** to modify the upper byte of the sp (stack pointer) register. This message is displayed as a status message only, since this action may not have been what you intended.

You wrote to "dirb" reg at 0f3 which may affect bank code reg

Bank switching is enabled, which means that the emulator has automatically set the 4 bank code bits in port B to digital outputs via the "dirb" register. You modified the dirb register using the **M**emory **M**odify command. The emulator displays this status message, since this action may not have been what you intended.

Zero wait state emulation does NOT support tram or trom

You selected **yes** for the configuration item "0 wait state execution?," and are trying to add map terms using **tram** or **trom** memory. The fast emulation memory mode doesn't support target system memory map terms, because the HPC mode that uses on-chip ROM doesn't support external memory accesses. Study your target system design and determine whether zero wait state operation is appropriate. Disable it if necessary by selecting **no** for the configuration item "0 wait state execution?"

9

HPC Emulator Syntax

Reference information specific to the HP 64775 emulator

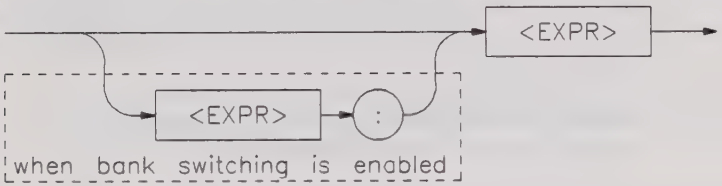
Topics Covered

- Syntax Diagrams and Variables
- Address Specification
- Configuration Items
- Demo command and demo program
- Mode
- Prompts
- Register Class

This chapter contains information about the commands and variables that are specific to the HPC emulator. Also included are descriptions of the syntax options and various examples of how to use the commands.

For information about emulation command syntax that is common to all HP 64700 Series emulators, see the *PC Interface Reference*.

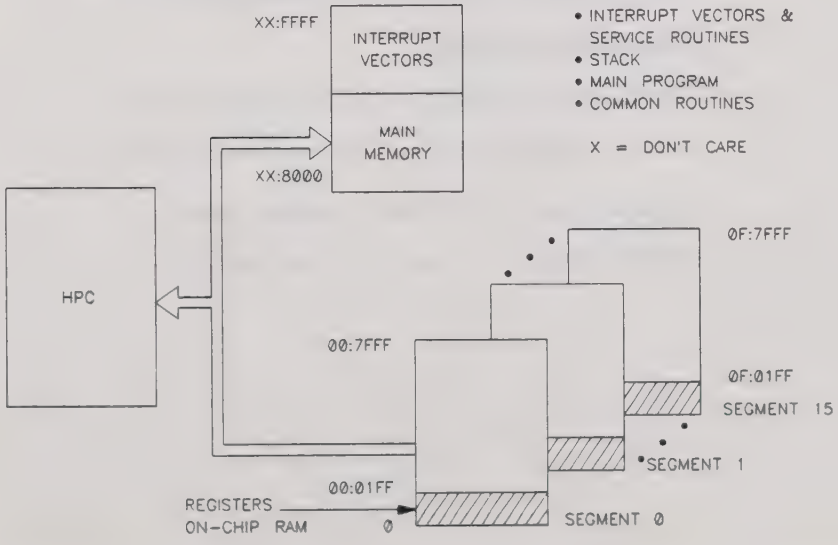
ADDRESS



The address expression (EXPR) allows you to enter an address in a form recognized by the HPC emulator. When you see the address variable in various syntax diagrams, remember that it is unique to the HPC emulator.

When bank switching is disabled, a valid HPC address is an expression that produces a 16 bit integer. (See chapter 4 for information on enabling/disabling bank switching.)

When bank switching is enabled, a valid HPC address can be an expression that produces a 16 bit integer. You can use this address form only for bank 0. A 4 bit integer, followed by a colon and a 16 bit integer specifies a full bank



address. The first integer specifies the bank, and the second integer specifies the offset within that bank.

In bank switched systems, you can refer to the area outside the bank switched range with an alias. For examples, the addresses 0:134, 1:134 and 0f:134 are equal, since they're in the on-chip RAM area. If the upper boundary of the bank switched range is 7fff hex, then the addresses 0:801a, 4:801a and 0e:801a are also equal.

When you don't specify a base, such as "y" for binary, "o" for octal, or "t" for decimal, the default is "h" for hexadecimal.

Examples

Bank Switching Disabled

The following command displays the contents of memory locations f000 through f00f hex.

```
Memory Display Bytes 0f000..0f00fh <Enter>
```

The following command displays the the code for the main portion of the emulator demo program in mnemonic format:

```
Memory Display Mnemonic INIT..END_MAIN <Enter>
```

Bank Switching Enabled

To display locations 500 through 50f hex in bank 1, enter:

```
Memory Display Bytes 1:500..1:50f <Enter>
```

To display location 7ffe in bank 10 decimal, enter:

```
Memory Display Bytes 0a:7ffe <Enter>
```

demo

The PC Interface provides a simple demo program for the HPC emulator that allows you to learn about the interface without the bother of writing a program.

The program emulates a simple command interpreter. It has a one-byte input buffer for commands, and recognizes the ASCII characters "A" and "B." All other values are considered invalid.

When you input a command to the buffer, the program copies it to a temporary location and sets a flag indicating that a command is pending. It then clears the input buffer.

Timer 2 is used to generate interrupts for processing the commands. When the program begins, the timer is initialized to provide an interrupt about every 16.7 ms. The interrupt service routine checks the pending command flag. If it's clear, the routine returns to the main program loop. Otherwise, it interprets the command and copies the appropriate message from a storage area to an output buffer.

There are two modules in the program. One is the main module, called **demo.asm**. The second module, which has the interrupt service routine, is called **int_serv.asm**. These files are listed here.

```
; HPC Demo Program
;
; This program provides a simple way to demonstrate the emulation and
; analysis capabilities of the HP 64775 emulator for the National HPC
; series of microcontrollers.
;
; The program simulates a primitive command interpreter. A location is
; designated as a command input location. This location is continually
; read by the input loop. If the location is non-zero, the contents
; are copied to a pending command location and a flag is set.
;
; Timer 2 of the HPC is used as a command processing timer. When Timer
; 2 underflows, it generates an interrupt. The interrupt service
; routine checks to see if a command is pending; if so, it interprets
; the command and writes the appropriate message to an output buffer
; area.

; Register name definitions

     enir=0x00d0:B
     r2=0x0186:W
     t2=0x0188:W
     tmmode=0x0190:W
     pwmode=0x0150:W
     divby=0x018e:W

; Export definitions and globals

     .public     enir,r2,t2,tmmode,pwmode,divby

     .public     CMD_PEND,PENDING_FLAG,MSG_A,MSG_B,MSG_I
     .public     END_MSGS,MSG_DEST,CMD_INPUT

;Import external variables

     .extrn     HANDLE_CMD

; Define interrupt vector for Timer interrupt

     .SECT     INTVECTSECT, ROM8

TIMERS=5
VECTOR:          .IPT     TIMERS, HANDLE_CMD

; Set section

     .SECT     DATABUFF, RAM8

; Define command input byte

CMD_INPUT:     .DSB     1

; Define command pending byte and command pending flag

CMD_PEND:     .DSB     1
PENDING_FLAG:     .DSB     1

; Define output buffer area
```

demo.asm

```
MSG_DEST:      .DSB     32

; New section for messages

     .SECT     MSGBUFF, ROM8

; Define message storage areas

MSG_A:          .BYTE     'Command A entered '
MSG_B:          .BYTE     'Entered B command '
MSG_I:          .BYTE     'Invalid command '

END_MSGS:       .BYTE     0


; Define section for subroutines

     .SECT     SUBROUTINES, ROM8


; The following subroutine clears the timer interrupt during timer
; interrupt servicing.

; Disable interrupts

     .public STOP_TIMER

STOP_TIMER:     LD     enir, #0

; Stop and reset T0..T3

          LD     tmmode, #0x4440
          NOP
          NOP
          LD     tmmode, #0xccc8
          RET

; The following subroutine restarts the timer and enables interrupts
; after timer interrupt servicing.

; Enable interrupts

     .public     START_TIMER

START_TIMER:    LD     enir, #0x21

; Start timer 2

          LD     tmmode, #0xc9c8
          RET

; Set program section for main program

          .SECT     MAINSECT, ROM8

; Initialization code: set up stack and timer, clear pending flag
```

demo.asm (Cont'd)

```
     .public     INIT

INIT:          LD     SP, #2
          LD     enir, #0      ; Disable interrupts
          LD     PENDING_FLAG, #0

; Set up timers per HPC user's manual
; Stop and reset t0..t3

          LD     tmmode, #0x4440
          NOP
          NOP
          LD     tmmode, #0xccc8

; Stop and reset t4-t7

          LD     pwmode, #0x4444
               ; no effect for non-003 applications
          NOP
          NOP
          LD     pwmode, #0xcccc
               ; no effect for non-003 applications

; The following should give interrupts about every 16.7 mS
; (60 Hz interrupt frequency) with a CKI of 20 MHz.

; Select divide by 131072 clock for t3, uart and uwire, divide by 32
; for t2.

          LD     divby, #0xf3ff

; Load Timer 2 and its input register

          LD     t2, #10437
          LD     r2, #10437

; Enable interrupts

          LD     enir, #0x21

; Start timer 2

          LD     tmmode, #0xc9c8

; Start of main code

; Clear command input byte

CLEAR_INPUT:    LD     CMD_INPUT, #0

; Read command input byte. If no command has been entered, continue
; to scan for input.

SCAN:          IFEQ     CMD_INPUT, #0
               JP     SCAN

; A command was entered. Copy it to the command pending byte, set the
; pending flag and return to CLEAR_INPUT label.
```

demo.asm (Cont'd)

```
PROCESS_CMD:     LD      CMD_PEND, CMD_INPUT
                 LD      CMD_INPUT, #0
                 LD      PENDING_FLAG, #1
                 JP      SCAN

END_MAIN:      .END      INIT
```

demo.asm (Cont'd)

```
; HPC Demo Program, Timer Interrupt Service Routine

; Import register name definitions

     .extrn      enir:B
     .extrn      r2:W
     .extrn      t2:W
     .extrn      tmmode:W
     .extrn      pwmode:W
     .extrn      divby:W

; Import global variables

     .extrn      STOP_TIMER:ROM8
     .extrn      START_TIMER:ROM8
     .extrn      CMD_PEND:RAM8:B
     .extrn      PENDING_FLAG:RAM8:B
     .extrn      MSG_A:ROM8:B
     .extrn      MSG_B:ROM8:B
     .extrn      MSG_I:ROM8:B
     .extrn      END_MSGS:ROM8:B
     .extrn      MSG_DEST:RAM8:B


; Define new section for interrupt handler

     .SECT      INTHNDLRSECT, ROM8

; Here's where the work is done. This is the interrupt service routine
; for the Timer 2 interrupt. First, check to see if there is a command
; pending. If not, return from interrupt. Otherwise, we skip the RETI
; and process the command.

     .public     HANDLE_CMD

HANDLE_CMD:      JSR      STOP_TIMER
                 IFEQ      PENDING_FLAG, #0
                 JP      NO_CMD
                 JP      INTERPRET_CMD

; There was no command.

NO_CMD:          JSR      START_TIMER
                 RETI

; The pending flag was non-zero, so we need to interpret the command.
```

int_serv.asm

```
INTERPRET_CMD:  IFEQ     CMD_PEND, #'A'
                JP       CMD_A
                IFEQ     CMD_PEND, #'B'
                JP       CMD_B
                JP       CMD_I

; We got command A, so set up the pointer to the beginning of that
; message.

CMD_A:          LD       X, #MSG_A
                LD       A, #MSG_B-MSG_A-1
                JP       WRITE_MSG

CMD_B:          LD       X, #MSG_B
                LD       A, #MSG_I-MSG_B-1
                JP       WRITE_MSG

CMD_I:          LD       X, #MSG_I
                LD       A, #END_MSGS-MSG_I-1
                JP       WRITE_MSG

; Write the message to the buffer.

WRITE_MSG:      LD       B, #MSG_DEST
                LD       K, B
                ADD      K, A

MSG_LOOP:       LD       A, [X+].B
                XS       A, [B+].B
                JP       MSG_LOOP

; Fill the remainder of the destination area with spaces (hex 20).

FILL_DEST:      LD       K, #MSG_DEST+32

FILL_LOOP:      LD       A, #0x20
                XS       A, [B+].B
                JP       FILL_LOOP

; Clean up by clearing the pending command, and reset the pending flag.

RESET_PEND:     LD       CMD_PEND, #0
                LD       PENDING_FLAG, #0


; Return from interrupt

                JSR      START_TIMER
                RETI

; End of interrupt handler
```

int_serv.asm (Cont'd)

EQUATES

Some status equates are predefined by the HPC emulator. These are:

Name	Value	Description
bgd	1xxxxxxxxxxy	Background memory cycle.
ea	0xx1xxxxxxxxy	State of the EA bit in the PSW (Program Status Word).
fci	0xxxxxxx11xxxy	ST2 status of "first cycle of instruction."
fgd	11111111111111111110xxxxxxxxxxy	Foreground memory cycle.
grd_ac	0xxxx1xxxxxxxy	Guarded memory access.
hbe	0xxxxxxxxxx1xy	State of the HPC's $\overline{\text{HBE}}$ pin.
hlda	0xxxxx1xxxxxxy	Hold Acknowledge asserted.
int	0xxxxxxx10xxxy	ST2 status of "interrupt."
norm	0xxxxxxx00xxxy	ST2 status of "normal."
read	0xxxxxxxxxxx0y	Read cycle.
reset	0xxxxxx1xxxxxy	Emulator or target system reset processor.
rom_wr	0xxx1xxxxxxxxy	Write to ROM.
skip	0xxxxxxx01xxxy	ST2 status of "skip."
st1	0xxxxxxxxx1xxy	First byte or opcode byte of instruction.
wo	0x0xxxxxxxxxxy	Watchdog output true.
write	0xxxxxxxxxx1y	Write cycle.

REGISTERS

Register Class	Register Names
basic	pc, a, b, k, x, sp, psw, fp, bc, locz
tmr0	t0con, tmmode, divby, r3, t3, r2, t2, i2cr, i3cr, i4cr
tmr4	eicr, eicon, portp, pwmode, r7, t7, r6, t6, r5, t5, r4, t4
uart	tbuf, rbuf, enur, enui, enu, divby
upi	upic, ibuf, obuf, portb, dirb, bfun
ints	irpd, enir, ircd, porti, psw, eicon, halten
adc	adr7, adr6, adr5, adr4, adr3, adr2, adr1, adr0, adcr3, adcr2, adcr1, portd
ht1	dmt1bc2, dmt1fa2, dmt1bc1, dmt1fa1, dmt1tc, portd, msgctl, msgpnd, serdec, msgcfg, hd1ctl, hd1est
ht2	dmt2bc2, dmt2fa2, dmt2bc1, dmt2fa1, dmt2tc, portd, msgctl, msgpnd, serdec, msgcfg, hd2ctl, hd2est
hr1	dmr1fl, dmr1da2, dmr1ca2, dmr1da1, dmr1ca1, dmr1rc, portd, msgctl, msgpnd, serdec, msgcfg, hd1crc2, hd1crc1, hd1est, hd1rst, hd1ctl, hd1ra2, hd1ra1
hr2	dmr2fl, dmr2da2, dmr2ca2, dmr2da1, dmr2ca1, dmr2rc, portd, msgctl, msgpnd, serdec, msgcfg, hd2crc2, hd2crc1, hd2est, hd2rst, hd2ctl, hd2ra2, hd2ra1
400e	portri, portro, dirr, udiv, udivr
uwire	sio, bfun, dirb, ircd, irpd, divby
ports	porta, portb, dira, dirb, bfun, portd, porti, portp

The register names correspond to HPC processor registers and memory locations as shown on the following pages.

HP 64775G (HPC 16003) Register Map

Emulator Register Name	Processor Register Name	Memory Address
locz	location zero	0001:0000
fp	frame pointer	00BF:00BE
psw	PSW Register	00C0
	(reserved)	00C1
	(reserved)	00C3:00C2
sp	SP Register	00C5:00C4
pc	PC Register	00C7:00C6
a	A Register	00C9:00C8
k	K Register	00CB:00CA
b	B Register	00CD:00CC
x	X Register	00CF:00CE
enir	ENIR Register	00D0
irpd	IRPD Register	00D2
ircd	IRCD Register	00D4
sio	SIO Register	00D6
porti	Port I Input Register	00D8
halten	HALT Enable Register	00DD:00DC
	Microcode ROM Dump	00DE
porta/obuf	Port A/OBUF	00E1:00E0
portb	Port B	00E3:00E2
bc	Bank Code Register	Bits 15, 12, 11, 10 of 00E3:00E2 when bank switching is enabled
upic	UPIC Register	00E6
dira/ibuf	DIR A Register/IBUF	00F1:00F0
dirb	DIR B Register	00F3:00F2
bfun	BFUN Register	00F5:00F4
portd	Port D Input Register	0104
enu	ENU Register	0120
enui	ENUI Register	0122
rbuf	RBUF Register	0124
tbuf	TBUF Register	0126

Emulator Register Name	Processor Register Name	Memory Address
enur	ENUR Register	0128
t4	T4 Timer	0141:0140
r4	R4 Register	0143:0142
t5	T5 Timer	0145:0144
r5	R5 Register	0147:0146
t6	T6 Timer	0149:0148
r6	R6 Register	014B:014A
t7	T7 Timer	014D:014C
r7	R7 Register	014F:014E
pwmode	PWMODE Register	0151:0150
portp	Port P Register	0153:0152
eicon	EICON	015C
eicr	EICR	015F:015E
i4cr	I4CR Register	0181:0180
i3cr	I3CR Register/T1	0183:0182
i2cr	I2CR Register/R1	0185:0184
r2	R2 Register	0187:0186
t2	T2 Timer	0189:0188
r3	R3 Register	018B:018A
t3	T3 Timer	018D:018C
divby	DIVBY Register	018F:018E
tmmode	TMMODE Register	0191:0190
t0con	T0CON Register	0192
wdreg	Watchdog Address	0195:0194

HP 64775H (HPC 16004) Register Map

Emulator Register Name	Processor Register Name	Memory Address
locz	location zero	0001:0000
fp	frame pointer	00BF:00BE
psw	PSW Register	00C0
	(reserved)	00C1
	(reserved)	00C3:00C2
sp	SP Register	00C5:00C4
pc	PC Register	00C7:00C6
a	A Register	00C9:00C8
k	K Register	00CB:00CA
b	B Register	00CD:00CC
x	X Register	00CF:00CE
enir	ENIR Register	00D0
irpd	IRPD Register	00D2
ircd	IRCD Register	00D4
sio	SIO Register	00D6
porti	Port I Input Register	00D8
halten	HALT Enable Register	00DD:00DC
	Microcode ROM Dump	00DE
porta/obuf	Port A/OBUF	00E1:00E0
portb	Port B	00E3:00E2
bc	Bank Code Register	Bits 15, 12, 11, 10 of 00E3:00E2 when bank switching is enabled
upic	UPIC Register	00E6
dira/ibuf	DIR A Register/IBUF	00F1:00F0
dirb	DIR B Register	00F3:00F2
bfun	BFUN Register	00F5:00F4
adcr1	A/D Control Register #1	0100
adcr2	A/D Control Register #2	0102
portd	Port D Input Register (A/D Analog Channel Inputs)	0104
adcr3	A/D Control Register #3	0106

Emulator Register Name	Processor Register Name	Memory Address
adr0	A/D Result Register 0	0111:0110
adr1	A/D Result Register 1	0113:0112
adr2	A/D Result Register 2	0115:0114
adr3	A/D Result Register 3	0117:0116
adr4	A/D Result Register 4	0119:0118
adr5	A/D Result Register 5	011B:011A
adr6	A/D Result Register 6	011D:011C
adr7	A/D Result Register 7	011F:011E
enu	ENU Register	0120
enui	ENUI Register	0122
rbuf	RBUF Register	0124
tbuf	TBUF Register	0126
enur	ENUR Register	0128
t4	T4 Timer	0141:0140
r4	R4 Register	0143:0142
t5	T5 Timer	0145:0144
r5	R5 Register	0147:0146
t6	T6 Timer	0149:0148
r6	R6 Register	014B:014A
t7	T7 Timer	014D:014C
r7	R7 Register	014F:014E
pwmode	PWMODE Register	0151:0150
portp	Port P Register	0153:0152
eicon	EICON	015C
eicr	EICR	015F:015E
i4cr	I4CR Register	0181:0180
i3cr	I3CR Register/T1	0183:0182
i2cr	I2CR Register/R1	0185:0184
r2	R2 Register	0187:0186
t2	T2 Timer	0189:0188
r3	R3 Register	018B:018A
t3	T3 Timer	018D:018C
divby	DIVBY Register	018F:018E

Emulator Register Name	Processor Register Name	Memory Address
tmmode	TMMODE Register	0191:0190
t0con	T0CON Register	0192
wdreg	Watchdog Address	0195:0194

HP 64775J (HPC 16400E) Register Map

Processor Register Group	Register Name	Emulator Register Name	Memory Address
	location zero	locz	0001:0000
	frame pointer	fp	00BF:00BE
HPC Core Registers	PSW Register	psw	00C0
	(reserved)		00C1
	(reserved)		00C3:00C2
	SP Register	sp	00C5:00C4
	PC Register	pc	00C7:00C6
	A Register	a	00C9:00C8
	K Register	k	00CB:00CA
	B Register	b	00CD:00CC
	X Register	x	00CF:00CE
Port Control and Interrupt Control Registers	ENIR Register	enir	00D0
	IRPD Register	irpd	00D2
	IRCD Register	ircd	00D4
	SIO Register	sio	00D6
	Port I Input Register	porti	00D8
	HALT Enable Register	halten	00DD:00DC
	Microcode ROM Dump		00DE
Port B	Port B Register	portb	00E3:00E2
	Bank Code Register	bc	Bits 15, 12, 11, 10 of 00E3:00E2 when bank switching is enabled
	Chip Revision Register	revreg	00E6
	Port B Direction Register	dirb	00F3:00F2
	BFUN Register	bfun	00F5:00F4
Ports R and D	Port D Input Register	portd	0100
	Message Control	msgctl	0102
	Message Pending	msgpnd	0104

Processor Register Group	Register Name	Emulator Register Name	Memory Address
	Serial Decoder Register	serdec	0106
	Message Configuration Register	msgcfg	0108
	Port R Output Register	portro	010A
	Port R Direction Register	dirr	010C
	Port R Input Register	portri	010E
UART	ENU Register	enu	0120
	ENUI Register	enui	0122
	RBUF Register	rbuf	0124
	TBUF Register	tbuf	0126
	ENUR Register	enur	0128
DMA 1 Receive	Receive Control & Status	dmr1rc	0141:0140
	Control Address 1	dmr1ca1	0143:0142
	Data Address 1	dmr1da1	0145:0144
	Control Address 2	dmr1ca2	0147:0146
	Data Address 2	dmr1da2	0149:0148
	Frame Length	dmr1fl	014B:014A
DMA1 Transmit	Transmit Control & Status	dmt1tc	0151:0150
	Field Address 1	dmt1fa1	0153:0152
	Byte Count 1	dmt1bc1	0155:0154
	Field Address 2	dmt1fa2	0157:0156
	Byte Count 2	dmt1bc2	0159:0158
DMA 2 Receive	Receive Control & Status	dmr2rc	0161:0160

Processor Register Group	Register Name	Emulator Register Name	Memory Address
	Control Address 1	dmr2ca1	0163:0162
	Data Address 1	dmr2da1	0165:0164
	Control Address 2	dmr2ca2	0167:0166
	Data Address 2	dmr2da2	0169:0168
	Frame Length	dmr2fl	016B:016A
DMA2 Transmit	Transmit Control & Status	dmt2tc	0171:0170
	Field Address 1	dmt2fa1	0173:0172
	Byte Count 1	dmt2bc1	0175:0174
	Field Address 2	dmt2fa2	0177:0176
	Byte Count 2	dmt2bc2	0179:0178
UART Timer	UART Reload		017D:017C
	UART Counter		017F:017E
Timer Block T0-T3	I4CR Register	i4cr	0181:0180
	I3CR Register/T1	i3cr	0183:0182
	I2CR Register/R1	i2cr	0185:0184
	R2 Register	r2	0187:0186
	T2 Timer	t2	0189:0188
	R3 Register	r3	018B:018A
	T3 Timer	t3	018D:018C
	DIVBY Register	divby	018F:018E
	TMMODE Register	tmmode	0191:0190
	T0CON Register	t0con	0192
Watchdog Logic	Watchdog Address	wdreg	0195:0194
HDLC #1	Receiver Address 1	hd1ra1	01A0

Processor Register Group	Register Name	Emulator Register Name	Memory Address
	Receiver Address 2	hd1ra2	01A2
	HDLC Control	hd1ctl	01A4
	Receiver Status	hd1rst	01A6
	Error Status	hd1est	01A8
	CRC 1st Byte Received	hd1crc1	01AA
	CRC 2nd Byte Received	hd1crc2	01AC
HDLC #2	Receiver Address 1	hd2ra1	01B0
	Receiver Address 2	hd2ra2	01B2
	HDLC Control	hd2ctl	01B4
	Receiver Status	hd2rst	01B6
	Error Status	hd2est	01B8
	CRC 1st Byte Received	hd1crc1	01BA
	CRC 2nd Byte Received	hd2crc2	01BC

The Register commands allow you to display and modify individual registers, or display groups of registers defined by a <REG_CLASS> identifier.

Examples

To display the basic set of registers, enter:

```
Register Display Class basic <Enter>
```

To display the HPC MicroWire registers, enter:

```
Register Display Class uwire <Enter>
```

To display the irpd register, enter:

```
Register Display Single irpd <Enter>
```

10

Specifications and Characteristics

Specifications and characteristics for the HP 64775 HPC emulator

General Emulator Specifications

Processor Compatibility

Compatible with National Semiconductor HPC processors and any processors meeting all the specifications of the HPC processors, according to the following:

Emulator Model Number	Processors Supported
HP 64775G	HPC16003/16083
HP 64775H	HPC16104/16164/16004/16064
HP 64775J	HPC16400E

Electrical

Maximum clock speed

The maximum internal and external clock speeds are as follows:

Emulator	Internal Clock	External Clock
HP 64775G/H	20 MHz, 0 wait states	20 MHz, 0 wait states 30 MHz, 1 wait state
HP 64775J	20 MHz, 1 wait state	20 MHz, 1 wait state

Minimum Clock Speed

Minimum external clock speed is 2 MHz for all supported processors.

Power

Power for oscillator chip on probe (74AC11000) drawn from target system. All other power supplied by the emulator.

Physical

Emulator Dimensions

173 mm height x 325 mm width x 389 mm depth (6.8 in. x 12.8 in. x 15.3 in.)

Emulator Weight

HP 64775, 8.2 kg (18 lb). (Any component used in suspending the emulator must be rated for 30 kg (65 lb) capacity.)

Cable Length

Emulator to target system, approximately 450 mm (1.5 ft).

Environmental

Temperature

Operating, 0° to +55° C (+32° to +131° F); nonoperating, -40° C to +70° C (-40° F to +158° F).

Altitude

Operating, 4600 m (15 000 ft), nonoperating, 15 300 m (50 000 ft).

Relative Humidity

15% to 95%.

Regulatory Compliance

Electromagnetic Interference

VDE 0871/6.78 Level A.

Safety Approvals

Self-certified to UL 1244, IEC 348, CSA 556B; meets IEC 380 for line isolation.

BNC (labeled TRIGGER IN/OUT)

Output Drive

Logic high level with 50 ohm load >= 2.0 V. Logic low level with 50 ohm load <= 0.4 V.

Input

74HCT132 with 135 ohms to ground in parallel. Maximum input: 5 V above Vcc; 5 V below ground.

Communications

Host Port

25-pin female type "D" subminiature connector.

RS-232-C DCE or DTE to 38.4 kbaud.

RS-422 DCE only to 460.8 kbaud.

Auxiliary Port

25-pin female type "D" subminiature connector.

RS-232-C DCE only to 19.2 kbaud.

CMB Port

9-pin female type "D" subminiature connector.

Index

A
ABG card
exchange, 167
absolute files
<file>.hpa created by HP 64000 Reader, 138
altitude specifications, 205
antistatic precautions, 155
ASCII symbol file (<file>.hps), 138

B
b command, 59
binary, 184

C
cautions
protect against static discharge, 73
turn OFF power before installing emulator probe, 73
verify pin 1 when installing emulator probe, 73
characteristics, 203
clock
specifications, 204
CMB
specifications, 206
commands
b, 59
s, 60

D
data communications specifications, 206
decimal, 184

E
electromagnetic interference specifications, 205
emulator probe
precautions, 74
equipment supplied, 155
exchange, 167

H
hexadecimal, 184
HP 64000 Reader command (RHP64000), 140

HP64000
Reader, using with PC Interface, 141
humidity specifications, 205

I installation
emulator, 157
overview, 156

L local symbols, 48, 140

M make file, 138
manuals
Terminal Interface Reference, 182

N note statements
date checking only in PC Interface, 142
use required options to include symbols, 138

O octal, 184

P parts list, 166 - 167
PC Interface
HP 64000 Reader, 141
power specifications, 204
probe cable length, 205

R registers
class variable, 192 - 201

S s command, 60
safety approvals, 205
single-step, 60
size specifications, 204
specifications
altitude, 205
clock, 204
CMB, 206
data communications, 206
dimensions, 204
electromagnetic interference, 205
humidity, 205
power, 204
probe cable, 205
safety approvals, 205

specifications (cont'd)
temperature, 205
timing, 166 - 167
trigger in/out, 206
weight, 205
static discharge, protecting the emulator probe against, 73
step, 60
symbols
.HPS file format, 139
local, 138
syntax diagrams
address variable, 183

T
temperature specifications, 205
timing specifications, 166 - 167
tools needed, 155
trigger
in/out specifications, 206

W
weight specifications, 205

HP 64775 Emulator PC Interface Manuals

Start Here

Concepts of Emulation and Analysis

Installation

HP 64700 Series Emulators Installation/Service Guide

Using the PC Interface

HP 64775 HPC Emulator PC Interface User's Guide

General Reference

HP 64700 PC Interface Reference

Certification and Warranty

Certification

Hewlett-Packard Company certifies that this product met its published specifications at the time of shipment from the factory. Hewlett-Packard further certifies that its calibration measurements are traceable to the United States National Bureau of Standards, to the extent allowed by the Bureau's calibration facility, and to the calibration facilities of other International Standards Organization members.

Warranty

This Hewlett-Packard system product is warranted against defects in materials and workmanship for a period of 90 days from date of installation. During the warranty period, HP will, at its option, either repair or replace products which prove to be defective.

Warranty service of this product will be performed at Buyer's facility at no charge within HP service travel areas. Outside HP service travel areas, warranty service will be performed at Buyer's facility only upon HP's prior agreement and Buyer shall pay HP's round trip travel expenses. In all other cases, products must be returned to a service facility designated by HP.

For products returned to HP for warranty service, Buyer shall prepay shipping charges to HP and HP shall pay shipping charges to return the product to Buyer. However, Buyer shall pay all shipping charges, duties, and taxes for products returned to HP from another country. HP warrants that its software and firmware designated by HP for use with an instrument will execute its programming instructions when properly installed on that instrument. HP does not warrant that the operation of the instrument, or software, or firmware will be uninterrupted or error free.

Limitation of Warranty

The foregoing warranty shall not apply to defects resulting from improper or inadequate maintenance by Buyer, Buyer-supplied software or interfacing, unauthorized modification or misuse, operation outside of the environment specifications for the product, or improper site preparation or maintenance.

No other warranty is expressed or implied. HP specifically disclaims the implied warranties of merchantability and fitness for a particular purpose.

Exclusive Remedies

The remedies provided herein are buyer's sole and exclusive remedies. HP shall not be liable for any direct, indirect, special, incidental, or consequential damages, whether based on contract, tort, or any other legal theory.

Product maintenance agreements and other customer assistance agreements are available for Hewlett-Packard products.

For any assistance, contact your nearest Hewlett-Packard Sales and Service Office.

Hewlett-Packard
Printed in the USA

SALES OFFICES
Arranged alphabetically by country

Please send directory corrections to:
Test & Measurement Catalog, 54-AF
Hewlett-Packard Company
5301 Stevens Creek Blvd.
P.O. Box 58059
Santa Clara, CA 95052-8059
Tel: (408) 553-7271
Fax: (408) 983-1006

HEADQUARTERS OFFICES

If no sales office is listed for your country or area, please contact one of these headquarters offices.

AUSTRALIA/ NEW ZEALAND

Hewlett-Packard Ltd.
31 - 41 Joseph St.
BLACKBURN, Victoria 3130
Tel: (61/3) 895-2895
Fax: (61/3) 898-7831
Telex: 31024MELB

FAR EAST

Hewlett-Packard Asia Ltd.
22/F Bond Centre, West Tower
89 Queensway, Central
G.P.O. Box 863
HONG KONG
Tel: (852/5) 848-7777
Fax: (852/5) 868-4997
Telex: 76793 HPA HX
Cable: HPASIAL TD

JAPAN

Yokogawa-Hewlett-Packard Ltd.
3-29-21, Takaido-Higashi
Suginami-Ku, **TOKYO** 168
Tel: (81/3) 331-6111
Fax: (81/3) 335-1478

UNITED KINGDOM

Hewlett-Packard Ltd.
Nine Mile Ride
WOKINGHAM
Berkshire, RG11 3LL
Tel: (44/34) 477-3100
Fax: (44/34) 476-3526

EASTERN EUROPE

Hewlett-Packard GmbH
Lieblgasse 1
P.O. Box 72
1222 **VIENNA,** Austria
Tel: (43/222) 2500-0
Fax: (43/222) 2500-444
Telex: 134 425

NORTHERN EUROPE

Hewlett-Packard S.A.
Van der Hooplaan 241
P.O. Box 999
1185 LN **AMSTELVEEN**
The Netherlands
Tel: (31/20) 547 9999
Fax: (31/20) 43 1120
Telex: 18919 hpner nl

SOUTH EAST EUROPE

Hewlett-Packard GmbH
Leiblgasse 1
P.O. Box 72
1222 **VIENNA,** Austria
Tel: (43/222) 2500-0
Fax: (43/222) 2500-444
Telex: 134 425 hpner nl

EUROPEAN MULTICOUNTRY REGION

Hewlett-Packard S.A.
150, Route du Nant d'Avril
1217 Meyrin 2
GENEVA, Switzerland
Tel: (41/22) 780 8111
Fax: (41/22) 780 8609

MIDDLE EAST AND AFRICA

Hewlett-Packard S.A.
International Sales Branch
Middle East/Africa
7, rue du Bois-du-Lan
P.O. Box 364
1217 Meyrin 1
GENEVA, Switzerland
Tel: (41/22) 780 7111
Fax: (41/22) 783 7535

CANADA

Hewlett-Packard (Canada) Ltd.
6877 Goreway Drive
MISSISSAUGA, Ontario L4V 1M8
Tel: (416) 678-9430
Fax: (416) 673-7253

LATIN AMERICA

Hewlett-Packard Company
Latin American Region, Miami Office
7208 N.W. 31st St.
MIAMI, FL 33122
United States
(305) 590-1400/1401
Fax: (305) 590-1427/1428
Telex: 441603HPMIAMI

UNITED STATES OF AMERICA

Customer Information Center
(800) 752-0900
6:00 AM to 5:00 PM Pacific Time

U.S. GOVERNMENT SPECIALISTS

ARMY
Tel: (201) 562-6107
Fax: (201) 562-6246
NAVY
Tel: (301) 258-2311
Fax: (301) 948-5986
AIR FORCE
Tel: (512) 491-1223
Fax: (512) 491-1299
FEDERAL PROGRAMS
Tel: (301) 362-7702
Fax: (301) 362-7650

EASTERN USA

Hewlett-Packard Co.
4 Choke Cherry Road
ROCKVILLE, MD 20850
Tel: (301) 670-4300

MIDWESTERN USA

Hewlett-Packard Co.
5201 Tollview Drive
ROLLING MEADOWS, IL 60008
Tel: (312) 255-9800

SOUTHERN USA

Hewlett-Packard Co.
2015 South Park Place
ATLANTA, GA 30339
Tel: (404) 955-1500

WESTERN USA

Hewlett-Packard Co.
5161 Lankershim Blvd.
NORTH HOLLYWOOD, CA 91601
Tel: (818) 505-5600

ALGERIA

Hewlett-Packard Trading S.A.
Bureau de Liaison
71, Lot. Benachour Abdelkader
CHERAGA
Tel: (213) 281 0287
Fax: (213) 281 0287
Telex: 0408 63161

ARGENTINA

Hewlett-Packard Argentina S.A.
Montaneses 2140/50
1428 BUENOS AIRES
Tel: (54/1) 781 4069
(54/1) 781-4090

AUSTRALIA

Brisbane, Queensland Office

Hewlett-Packard Australia Ltd.
10 Payne Road
THE GAP, Queensland 4061
Tel: (61/7) 300-4133
Fax: (61/7) 300-5592
Telex: 42133
Cable: HEWPARD Brisbane

Canberra, Australia Capital Territory Office

Hewlett-Packard Australia Ltd.
Thynne Street, Fern Hill Park
BRUCE, A.C.T. 2617
Tel: (61/6) 251-6999
Fax: (61/6) 251-6948
Telex: 62650
Cable: HEWPARD Canberra

Melbourne, Victoria Office

Hewlett-Packard Australia Ltd.
126-142 Trennery Crescent
Abbotsford, VIC 3067
Tel: (61/3) 895-2895
Fax: (61/3) 898-7831
Cable: HEWPARD Melbourne

Parkside, South Australia Office

Hewlett-Packard Australia Ltd.
153 Greenhill Road
PARKSIDE, S.A. 5063
Tel: (61/8) 272-5911
Fax: (61/8) 373-1398

Perth, Western Australia Office

Hewlett-Packard Australia Ltd.
Herdsman Business Park
Cnr. Hasler & Gould St.
OSBORNE PARK, W.A. 6017
Tel: (61/9) 441-8000
Fax: (61/9) 242-1682
Cable: HEWPARD Perth

Sydney, New South Wales Office

Hewlett-Packard Australia Ltd.
17-23 Talavera Road
P.O. Box 308
NORTH RYDE, N.S.W. 2113
Tel: (61/2) 888-4444
Fax: (61/2) 888-9072
Cable: HEWPARD Sydney

AUSTRIA

Hewlett-Packard GmbH
Verkaufsbüro Graz
Grottenhofstrasse 94
8052 **GRAZ**
Tel: (43/316) 28 3066
Fax: (43/316) 28 1756
Telex: 312375

Hewlett-Packard GmbH
Lieblgasse 1
P.O. Box 72
1222 **VIENNA**
Tel: (43/222) 2500-0
Fax: (43/222) 2500 - 444

BAHRAIN

Wael Pharmacy and Drugstore
P.O. Box 648
MANAMA
Tel: (973) 25 61 23
Fax: (973) 23 36 38
Telex: 8550 WAEL BN

SALES OFFICES
Arranged alphabetically by country (cont'd)

BARBADOS
Computers & Controls (Barbados) Ltd.
Suite 5, 1st Floor
Wildey Shopping Plaza
ST. MICHAEL
Tel: (809) 429-4103
Fax: (809) 429-4103

BELGIUM
Hewlett-Packard Belgium S.A./N.V.
Blvd de la Woluwe, 100
Woluwedal
1200 BRUSSELS
Tel: (32/2) 761 31 11
Fax: (32/2) 763 06 13

BENIN
Engineering Business Concept (E.B.C)
21, Route du Canal
Zone 3
04 B.P. 1357
ABIDJAN
Ivory Coast
Tel: (225) 21 50 24
Fax: (225) 35 37 90

BERMUDA
Applied Computer Technologies
17 Reid St.
HAMILTON 5
Tel: (809) 295-1616
Fax: (809) 292-7967
Telex: 380 3589/ACT BA

BOLIVIA
Siser Ltda. (Sistemas de
Importacion y Servicios Ltda.)
Gabriel Gozalvez 221
Casilla 4084
LA PAZ
Tel: (591/2) 340962/
363365/343245
Fax: (591/2) 359268

BRAZIL
Edisa informatica S.A.
Alameda Rio Negro, 750-Alphaville
06454-**BARUERI**-SP
Tel: (55/11) 709-1444
Fax: (55/11) 709-1244
Telex: 1171351

Edisa Informatica S.A.
Rua dos Andradas, 1001-10 andar
90020-**PORTO ALEGRE**-RS
Tel: (55/51) 225-7166
Fax: (55/51) 225-4060
Telex: 515663

Edisa Informatica S.A.
Av. Eng. Abdias de Carvalho, 1111-5 andar
50751-**RECIFE**-PE
Tel: (55/81) 227-2722
Fax: (55/81) 228-3793
Telex: 812904

Edisa Informatica S.A.
Praia de Botafogo, 228-4 andar
22250-**RIO DE JANEIRO**-RJ
Tel: (55/21) 552-0222
Fax: (55/21) 551-1449
Telex: 2121905

BRUNEI
Komputer Wisman Sdn Bhd
Room 1, 3rd Floor
Jalan Tutong
BS Begawan
BRUNEI DARUSSALEM
Tel: (673/2) 23918
Telex: 0809 2447

BURKINA FASSO
Engineering Business Concept (E.B.C.)
21, Route du Canal
04 B.P. 1357
ABIDJAN
Ivory Coast
Tel: (225) 21 50 24
Fax: (225) 35 37 90

CANADA

Alberta
Hewlett-Packard (Canada) Ltd.
150 - 6th Avenue S.W.
CALGARY, Alberta T2P 3Y7
Tel: (403) 262-0777
Fax: (403) 237-9309
Hewlett-Packard (Canada) Ltd.
11120-178th Street
EDMONTON, Alberta T5S 1P2
Tel: (403) 486-6666
Fax: (403) 489-8764

British Columbia
Hewlett-Packard (Canada) Ltd.
10691 Shellbridge Way
RICHMOND,
British Columbia V6X 2W8
Tel: (604) 270-2277
Fax: (604) 270-0859

Manitoba
Hewlett-Packard (Canada) Ltd.
1825 Inkster Blvd.
WINNIPEG, Manitoba R2X 1R3
Tel: (204) 694-2777
Fax: (204) 694-3901

Nova Scotia
Hewlett-Packard (Canada) Ltd.
201 Brownlow Avenue
DARTMOUTH, Nova Scotia B3B 1W2
Tel: (902) 468-4725
Fax: (902) 468-2817

Hewlett-Packard (Canada) Ltd.
475 Hood Rd., Unit #2
MARKHAM, Ontario L3R 8H1
Tel: (416) 479-1770
Fax: (416) 479-3105

Ontario
Hewlett-Packard (Canada) Ltd.
552 Newbold Street
LONDON, Ontario N6E 2S5
Tel: (519) 686-9181
Fax: (519) 686-9145

Hewlett-Packard (Canada) Ltd.
6877 Goreway Drive
MISSISSAUGA, Ontario L4V 1M8
Tel: (416) 678-9430
Fax: (416) 678-9421

Hewlett-Packard (Canada) Ltd.
2670 Queensview Dr.
OTTAWA, Ontario K2B 8K1
Tel: (613) 820-6483
Fax: (613) 820-0377

Quebec
Hewlett-Packard (Canada) Ltd.
17500 Trans Canada Highway
South Service Road
KIRKLAND, Quebec H9J 2X8
Tel: (514) 697-4232
Fax: (514) 697-6941

Saskatchewan
Hewlett-Packard (Canada) Ltd.
#1, 2175 Airport Rd.
SASKATOON, Saskatchewan S7L 7E1
Tel: (306) 242-3702

CHILE
Avanzados Sistemas de
Conocimientos S. A. (ASC)
Austria 2041
SANTIAGO
Tel: (56/2) 223-5946/6148
Fax: (56/2) 223-1912

CHINA, PEOPLE'S REPUBLIC OF
China Hewlett-Packard Co., Ltd.
38 Bei San Huan Xi Road
Shuang Yu Shu, Hai Dian District
P.O. Box 9610
BEIJING
Tel: (86/1) 256-6888
Fax: (86/1) 256-3207

China Hewlett-Packard Co., Ltd.
28/F Shanghai Union Building
100 Yan An Dong Road
SHANGHAI 200002
Tel: (86/21) 320-3240
Fax: (86/21) 320-2149

COLOMBIA
Carbajal S.A.
Calle 29 Norte No. 6 A 40
Apartado Aereo 46
CALI
Tel: (57/23) 675 011
Fax: (57/23) 688 466

COSTA RICA
I.S. de Costa Rica S.A.
Calle 25, Avs. 6 y 8 No.648
SAN JOSE
Tel: (50/6) 33 3722
Fax: (50/6) 55 3528

CYPRUS
Telerexa Ltd.
P.O. Box 1152
Valentine House
8 Stassandrou St.
NICOSIA
Tel: (357/2) 445 628
Telex: 0605 5845 tlrx cy

DENMARK
Hewlett-Packard A/S
Birkerød Kongevejen 25
3460 **BIRKERØD**
Tel: (45/42) 81 66 40
Fax: (45/42) 81 58 10

Hewlett-Packard A/S
Voldbjergvej 16
8240 **RISSKOV,** Aarhus
Tel: (45/06) 17 60 00
Fax: (45/06) 17 60 58
Telex: 37409 hpas dk

DOMINICAN REPUBLIC
Esacomp S.A.
Edificio Mercantil del Caribe
Ave. John F. Kennedy No. 11
P.O. Box 1496
SANTO DOMINGO
Tel: (809) 567 3241
Fax: (809) 566 9774
Telex: ITT 346 0439

ECUADOR
CYEDE Cia. Ltda.
Avenida Eloy Alfaro 1749
y Belgica
Casilla 6423 CCI
QUITO
Tel: (593) 245-0075
Fax: (593) 224-4223
Telex: 39322548 CYEDE ED

EGYPT
International Engineering Associates
6 El Gamea Street
AGUZA/CAIRO
Tel: (20/2) 71 21 68
Telex: 93830 IEA UN
Cable: INTEGASSO

EL SALVADOR
IPESA de El Salvador S.A.
29 Avenida Norte 1223
SAN SALVADOR
Tel: (503) 266 858
Telex: 301 20539 IPESA SAL

FINLAND
Hewlett-Packard Oy
Piispankalliontie 17
02200 **ESPOO (Helsinki)**
Tel: (358/0) 887 21
Fax: (358/0) 887 22 77

Hewlett-Packard Oy
Väinönkatu 9 C
40100 **JYVÄSKYLÄ**
Tel: (358/41) 21 85 11

Hewlett-Packard Oy
Valtatie 57
90500 **OULU**
Tel: (358/81) 340 144
Fax: (358/81) 340 145

FRANCE

Aix-En-Provence
Hewlett-Packard France
ZI Mercure B
Rue Berthelot
13763 **LES MILLES** Cédex
Tel: (33) 42 24 32 43
Fax: (33) 42 59 48 72
Telex: 410 770

Bordeaux
Hewlett-Packard France
Domaine de Pélus
5, avenue de Pythagore
33700 **MERIGNAC**
Tel: (33) 56 34 00 84
Fax: (33) 56 34 80 84
Telex: 550 105

Brest
Hewlett-Packard France
ZAC Kergaradec
8, rue Fernand Forest
29239 **GOUESNOU**
Tel: (33) 98 41 87 90
Fax: (33) 98 41 74 77

Grenoble
Hewlett-Packard France
57, chemin du Vieux Chêne
ZIRST
38240 **MEYLAN**
Tel: (33) 76 90 38 40
Fax: (33) 76 41 05 36

Lille
Hewlett-Packard France
Parc d'activités des Prés
1, rue Papin
59658 **VILLENEUVE D'ASCQ**
Tel: (33) 20 47 78 78
Fax: (33) 20 33 36 77
Telex: 160 124

Lyon
Hewlett-Packard France
Chemin des Mouilles
BP 162
69131 **ECULLY** Cédex
Tel: (33) 72 29 32 93
Fax: (33) 78 33 49 82
Telex: 310 617

Nice
Hewlett-Packard France
Les Cardoulines - Bât. 2
Route des Dolines
Sophia Antipolis
06560 **VALBONNE**
Tel: (33) 93 65 39 40
Fax: (33) 93 65 31 34

Orléans
Hewlett-Packard France
Parc Tertiaire Héliopolis
Route de Micy
45380 **LA CHAPELLE ST MESMIN**
Tel: (33) 38 43 94 56
Fax: (33) 38 88 22 81
Telex: 783 497

Orsay
Hewlett-Packard France
Zone Industrielle de Courtaboeuf
1, avenue du Canada
91947 **LES ULIS** Cédex
Tel: (33/1) 69 82 60 60
Fax: (33/1) 69 82 60 61
Telex: 600 048

Rennes
Hewlett-Packard France
Parc d'activités de la Poterie
Rue Louis Kerautret-Botmel
35000 **RENNES**
Tel: (33) 99 51 42 44
Fax: (33) 99 32 29 19
Telex: 740 912

Rouen
Hewlett-Packard France
PAT Lavatine
3, rue Jacques Monod
BP 228
76136 **MONT-ST-AIGNAN**
Tel: (33) 35 59 19 20
Fax: (33) 35 59 65 11
Telex: 770 035

Strasbourg
Hewlett-Packard France
4, rue de la Faisanderie
Parc Club des Tanneries
BP 40
67381 **LINGOLSHEIM** Cédex
Tel: (33) 88 76 15 00
Fax: (33) 88 78 17 63
Telex: 890 141

Toulouse
Hewlett-Packard France
Innoparc
BP 167 Vole n° 7
31328 **LABEGE** Cédex
Tel: (33) 61 39 11 40
Fax: (33) 61 39 10 97
Telex: 531 639

GERMAN FEDERAL REPUBLIC

HEADQUARTERS
Hewlett-Packard GmbH
Herrenberger Strasse 130
7030 **BÖBLINGEN**
Tel: (49/7031) 14-0
Fax: (49/7031) 14-2999

Hewlett-Packard GmbH
Bad Homburg
Hewlett-Packard-Strasse
6380 **BAD HOMBURG**
Tel: (49) 6172 16-0
Fax: (49) 6172 16-1309
Telex: 410 844 hpbhg

Berlin
Hewlett-Packard GmbH
Lützowplatz 15
1000 **BERLIN** 30
Tel: (49/30) 2500 02-0
Fax: (49/30) 2500 02-62
Telex: 018 3405 hpbln d

Böblingen
Hewlett-Packard GmbH
Schickardstrasse 2
7030 **BÖBLINGEN**
Tel: (49/7031) 645-0
Fax: (49/7031) 645-429

Bonn
Hewlett-Packard GmbH
Friedrich-Ebert-Allee 26
5300 **BONN** 1
Tel: (49/228) 23400-1
Fax: (49/228) 235 315

Dortmund
Hewlett-Packard GmbH
Schleefstrasse 28
4600 **DORTMUND**-Aplerbeck
Tel: (49/231) 45001-0
Fax: (49/231) 45001-37
Telex: 822858 hepdod

Frankfurt
Hewlett-Packard GmbH
Reparaturzentrum Frankfurt
Berner Strasse 117
6000 **FRANKFURT 56**
Tel: (49/69) 5000 60
Fax: (49/69) 5000 6-200
Telex: 412 815 hpffm

Hamburg
Hewlett-Packard GmbH
Kapstadtring 5
2000 **HAMBURG 60**
Tel: (49/40) 63804-0
Fax: (49/40) 63804-327
Telex: 2163032

Hannover
Hewlett-Packard GmbH
Heidering 37-39
3000 **HANNOVER** 61
Tel: (49/511) 5706-0
Fax: (49/511) 5706-126

Mannheim
Hewlett-Packard GmbH
Rosslauer Weg 2-4
6800 **MANNHEIM** 31
Tel: (49/621) 7005-0
Fax: (49/621) 7005-200
Telex: 462105

Munchen (Munich)
Hewlett-Packard GmbH
Eschenstrasse 5
8028 **TAUFKIRCHEN**
Tel: (49/89) 61207-0
Fax: (49/89) 61207-300
Telex: 0524985 hpmch

Nürnberg
Hewlett-Packard GmbH
Emmericher Strasse 13
8500 **NÜRNBERG** 10
Tel: (49/911) 5205-0
Fax: (49/911) 5205-140
Telex: 623860

Ratingen
Hewlett-Packard GmbH
Berliner Strasse 111
4030 **RATINGEN** 1
Tel: (49/2102) 494-0
Fax: (49/2102) 494-300
Telex: 8589070

Ulm
Hewlett-Packard GmbH
Messerschmittstrasse 7
7910 **NEU ULM**
Tel: (49/731) 7073-0
Fax: (49/731) 7076-66
Telex: 712816

Waldbronn
Hewlett-Packard GmbH
Ermlis-Allee
7517 **WALDBRONN** 2 (Karlsruhe)
Tel: (49/7243) 602-0
Fax: (49/7243) 602-512
Telex: 7265743

GHANA
Engineering Business Concept (E.B.C.)
21, Route du Canal
Zone 3
04 B.P. 1357
ABIDJAN
Ivory Coast
Tel: (225) 31 50 24
Fax: (225) 35 37 90

GREAT BRITAIN
See United Kingdom

GREECE
Hewlett-Packard Hellas
32, Kiffissias Avenue
15125 Amaroussion
ATHENS
Greece
Tel: (30/1) 682 88 11
Fax: (30/1) 683 29 78
Telex: 216588 hpat gr

GUATEMALA
Ipesa de Guatemala
Avenida Reforma 3-48, Zona 9
GUATEMALA CITY
Tel: (502) 231-7853
Fax: (502) 231-6627
Telex: 3055765 IPESA GU

HONG KONG
Hewlett-Packard Asia, Ltd.
22/F, Bond Centre, West Tower
89 Queensway Central
HONG KONG
Tel: (852/5) 848-7777
Fax: (852/5) 868-4997
Cable: HEWPACK HONG KONG

SALES OFFICES
Arranged alphabetically by country (cont'd)

HUNGARY
Hewlett-Packard
Accredite Office Hungary
Radvany u. 7
1118 **BUDAPEST**
Tel: (36/1) 185 23 68
Fax: (36/1) 165 10 85
Telex: 861 227 632

ICELAND
Hewlett-Packard Iceland
Høfdabakka 9
110 **REYKJAVIK**
Tel: (354/1) 67-1000
Fax: (354/1) 67-3031
Telex: 37409

INDIA

Bangalore
Hewlett-Packard India Pvt. Ltd.
29 Cunningham Road
BANGALORE 560 052
Tel: (91/812) 261075
Fax: (91/812) 261554

Bombay
Hewlett-Packard India Pvt. Ltd.
Sahas
414/2 Veer Savarkar Marg
Prabhadevi
BOMBAY 400 025
Tel: (91/22) 4306155
Fax: (91/22) 4307078

Calcutta
Hewlett-Packard India Pvt. Ltd.
DBS Executive Centre
8 Acharya J. C. Bose Road
CALCUTTA 700 017
Tel: (91/33) 444990
Fax: (91/33) 444614

Hyderabad
Hewlett-Packard India Pvt. Ltd.
9-5-13 Taramandal Complex
9th Floor
Saifabad
HYDERABAD 500 004
Tel: (91/842) 231756
Fax: (91/842) 8313444

New Delhi
Hewlett-Packard India Pvt. Ltd.
B-8 Jangpura
Mathura Road
NEW DELHI 110 014
Tel: (91/11) 690329
Fax: (91/11) 353315

INDONESIA
BERCA Indonesia P.T.
Wisma Dharmala Sakti
10/F, J1. Jendral Sudirman K.av 32
P.O. Box 41/JKPDS
JAKARTA 10001
Tel: (62/21) 578-0005
Fax: (62/21) 570-1287
Telex: 62065 BERCAMIA

IRAQ
Hewlett-Packard Trading S.A.
(Service Operation)
Al Mansoor City 609/10/7
BAGHDAD
Tel: (964/1) 541 49 73
Fax: (964/1) 541 49 73
Telex: 212455 hepairagik

IRELAND
Hewlett-Packard Ireland Ltd.
Temple House, Temple Road
Blackrock, Co. **DUBLIN**
Tel: (353/1) 883399
Fax: (353/1) 883742
Telex: 30439

ISRAEL
Computation and Measurement
Systems (CMS) Ltd.
11, Hashlosha Street
TEL-AVIV 67060
Tel: (972) 3 5380-333
Fax: (972) 3 5375-055
Telex: 371234 HPCMS

ITALY

Anzola Emilia-Bologna
Hewlett-Packard Italiana S.p.A.
Via Emilia, 51/C
40011 **ANZOLA EMILIA-BOLOGNA**
Tel: (39/51) 73 10 61
Fax: (39/51) 73 48 30
Telex: 51 16 30

Bari
Hewlett-Packard Italiana S.p.A.
Via Vitantonio di Cagno, 34
70124 **BARI**
Tel: (39/80) 41 07 44
Fax: (39/80) 41 78 51

Catania
Hewlett-Packard Italiana S.p.A.
Via Principe Nicola, 43 G/C
95126 **CATANIA**
Tel: (39/95) 37 10 87
Fax: (39/95) 38 85 69

Corsico
Hewlett-Packard Italiana S.p.A.
Via G. di Vittorio, 10
20094 **CORSICO** (MI)
Tel: (39/2) 440 83 51
Fax: (39/2) 440 95 64
Telex: 440 9564

Firenze
Hewlett-Packard Italiana S.p.A.
Via Sacco e Vanzetti, 1
50145 **FIRENZE**
Tel: (39/55) 31 85 53
Fax: (39/55) 37 39 65

Genova
Hewlett-Packard Italiana S.p.A.
Viale Brigata Bisagno, 2
16129 **GENOVA**
Tel: (39/10) 54 11 41
Fax: (39/10) 59 17 33
Telex: 28 52 38

Limito
Hewlett-Packard Italiana S.p.A.
Via Nuova Rivoltana, 95
20090 **LIMITO** (MI)
Tel: (39/2) 757 61
Fax: (39/2) 757 6230
Telex: 32 31 16

Napoli
Hewlett-Packard Italiana S.p.A.
Via Orazio, 16
80122 **NAPOLI**
Tel: (39/81) 761 14 44
Fax: (39/81) 68 01 64
Telex: 71 06 98

Padova
Hewlett-Packard Italiana S.p.A.
Via Pellizzo, 15
35128 **PADOVA**
Tel: (39/49) 807 01 66
Fax: (39/49) 77 30 97
Telex: 43 03 15

Roma-Eur
Hewlett-Packard Italiana S.p.A.
Via del Tintoretto, 200
00142 **ROMA-EUR**
Tel: (39/6) 5 48 31
Fax: (39/6) 540 87 10
Telex: 62 65 24

Torino
Hewlett-Packard Italiana S.p.A.
C.so Svizzera, 185
10149 **TORINO**
Tel: (39/11) 74 40 44
Fax: (39/11) 771 08 15
Telex: 22 10 79

IVORY COAST
Engineering Business Concept (E.B.C.)
Angle Avenue J. Anoma et Bd.
République
08 B.P. 323 **ABIDJAN** 08
Tel: (225) 32 50 24/41 48 70
Fax: (225) 35 37 90

JAPAN

Akita
Yokogawa-Hewlett-Packard Ltd.
Nihonseimei Akita Chudori Bldg.
4-2-7 Nakadori
AKITA 010
Tel: (81/188) 36-5021
Fax: (81/188) 36-5099

Atsugi
Yokogawa-Hewlett-Packard Ltd.
Keno Kikaku Kogyo No. 2 Bldg.
9-32 Tamuracho
ATSUGI 243
Tel: (81/462) 25-0031
Fax: (81/462) 25-0064

Chiba
Yokogawa-Hewlett-Packard Ltd.
Fujimoto Daiichi Seimei Bldg.
3-3-1 Chuo
CHIBA 280
Tel: (81/472) 25-7701
Fax: (81/472) 21-0382

Fukuoka
Yokogawa-Hewlett-Packard Ltd.
Daisan Hakata-Kaisei Bldg.
1-3-6 Hakataekiminami Hakata-ku
FUKUOKA 812
Tel: (81/92) 472-8731
Fax: (81/92) 473-4745

Hiroshima
Yokogawa-Hewlett-Packard Ltd.
Yasuda-semei Hiroshima Bldg.
6-11 Hondori Naka-ku
HIROSHIMA 730
Tel: (81/82) 241-0611
Fax: (81/82) 241-0619

Kobe
Yokogawa-Hewlett-Packard Ltd.
Towa Building
2-2-3 Kaigandori, Chuo-ku
KOBE 650
Tel: (81/78) 392-4791
Fax: (81/78) 392-4839

Koriyama
Yokogawa-Hewlett-Packard Ltd.
Nihon-Dantai-seimei Bldg.
21-10 Toramaru-machi
KORIYAMA 963
Tel: (81/249) 39-7111
Fax: (81/249) 39-7234

Kumagaya
Yokogawa-Hewlett-Packard Ltd.
Kumagaya Asahi 82 Bldg.
3-4 Tsukuba
KUMAGAYA 360
Tel: (81/485) 24-6563
Fax: (81/485) 24-9050

Kyoto
Yokogawa-Hewlett-Packard Ltd.
Shin-Kyoto Center Bldg.
614 Higashi-Shokoji-cho
Karasuma-nishi-iru, Shiokoji-dori
Shimogyo-ku
KYOTO 600
Tel: (81/75) 343-0921
Fax: (81/75) 343-4356

Mito
Yokogawa-Hewlett-Packard Ltd.
Mito Mitsui Bldg.
1-4-73 Sannomaru
MITO 310
Tel: (81/292) 25-7470
Fax: (81/292) 31-6589

Miyazaki
Yokogawa-Hewlett-Packard Ltd.
Nomura Shoken Bldg.
4-1-2 Tachibana-dori
Miyazaki-C.
MIYAZAKI 880
Tel: (81/985) 23-7280
Fax: (81/985) 23-7864

Nagano
Yokogawa-Hewlett-Packard Ltd.
Nagano-Tokyo-kaijo Bldg.
1081, Minamiagatamachi
NAGANO 380
Tel: (81/262) 24-8012
Fax: (81/262) 24-8016

Nagoya
Yokogawa-Hewlett-Packard Ltd.
Nagoya Kokusai Center Bldg.
1-47-1 Nakono, Nakamura-ku
NAGOYA 450
Tel: (81/52) 571-2262
Fax: (81/52) 565-0896

Nara
Yokogawa-Hewlett-Packard Ltd.
Nissei Nara Ekimae Bldg.
1-1-15 Omiya-cho
NARA 630
Tel: (81/742) 22-8235
Fax: (81/742) 22-8219

Oomiya
Yokogawa-Hewlett-Packard Ltd.
Saikyoren Bldg.
1-2 Dotemachi
OOMIYA 330
Tel: (81/48) 645-8031
Fax: (81/48) 645-8001

Osaka
Yokogawa-Hewlett-Packard Ltd.
Chuo Bldg.
5-4-20 Nishinakajima
Yodogawa-ku
OSAKA 532
Tel: (81/6) 304-6021
Fax: (81/6) 304-0216

Sapporo
Yokogawa-Hewlett-Packard Ltd.
Sumitomo-seimei Sapporo Chu Bldg.
1-1-14 Nanjo-higashi Chuo-ku
SAPPORO 060
Tel: (81/11) 251-1011
Fax: (81/11) 222-3239

Sendai
Yokogawa-Hewlett-Packard Ltd.
Yamaguchi Bldg.
1-1-31 Ichibancho Aoba-ku
SENDAI 980
Tel: (81/22) 225-1011
Fax: (81/22) 225-1616

Suwa-gun
Yokogawa-Hewlett-Packard Ltd.
Daimon Bldg.
139-1 Yaginishi Shimosuwa-cho
SUWA-GUN 393
Tel: (81/266) 28-8851
Fax: (81/266) 28-8873

Tokyo
Yokogawa-Hewlett-Packard Ltd.
Shinjuku Daiichi-seimei Bldg.
2-7-1 Nishi-shinjuku
Shinjuku-ku
TOKYO 160
Tel: (81/3) 342-2734
Fax: (81/3) 348-7969

Yokogawa-Hewlett-Packard Ltd.
9-1 Takakura-cho
Hachuoji-shi
TOKYO 192
Tel: (81/426) 42-1231
Fax: (81/426) 45-2631

Yokogawa-Hewlett-Packard Ltd.
Tokyo-Nissan Bldg.
1-6-34 Konan, Minato-ku
TOKYO, 108
Tel: (81/3) 458-5411
Fax: (81/3) 458-5400

Yokogawa-Hewlett-Packard Ltd.
3-29-21 Takaido-higashi
Suginami-ku
TOKYO 168
Tel: (81/3) 331-6111
Telex: 232 2024 YHPTOK

Yokogawa-Hewlett-Packard Ltd.
Tokyu Sakuragaokacho Bldg.
31-2 Sakuragaoka Shibuya-ku
TOKYO 150
Tel: (81/3) 780-5511
Fax: (81/3) 780-5510

Toyota
Yokogawa-Hewlett-Packard Ltd.
Toyota-Tokyo-Kaijo Bldg.
1-179 Miyuki-honmachi
TOYOTA 473
Tel: (81/565) 27-5611
Fax: (81/565) 27-5400

Tsukuba
Yokogawa-Hewlett-Packard Ltd.
Issei Bldg.
2-3-17 Takezono
TSUKUBA 305
Tel: (81/298) 51-5141
Fax: (81/298) 51-5381

Utsunomiya
Yokogawa-Hewlett-Packard Ltd.
Chiyodaseimei-Utsunomiya Bldg.
2-3-1 Odori
UTSUNOMIYA 320
Tel: (81/286) 33-1153
Fax: (81/286) 33-1175

Yokohama
Yokogawa-Hewlett-Packard Ltd.
No. 2 Yasuda Bldg.
2-32-12 Tsuruyacho
Kanagawa-ku
YOKOHAMA-SHI KANAGAWA
Tel: (81/45) 313-1352
Fax: (81/45) 312-1846

JORDAN
Scientific & Medical Supplies Co.
P.O. Box 1387
AMMAN
Tel: (962/6) 62 49 07
Fax: (962/6) 62 82 58
Telex: 0493 21456 sabco jo

KENYA
ADCOM Ltd., Inc., Kenya
P.O. Box 30070
NAIROBI
Tel: (254/2) 33 1955
Telex. 22639

KOREA
Samsung Hewlett-Packard Co. Ltd.
Dongbang Building, 12-16/F
36-1 Yeoeuido-dong
Youngdeungpo-Ku
SEOUL, 150
Tel: (82/2) 769-0114
Fax: (82/2) 784-7084
Telex: SAM HP K 25166

Samsung Hewlett-Packard Co. Ltd.
13/F Taegu Bank Bldg.
118, 2-Ka
Suseong-Dong
Suseong-Ku
TAEGU
Tel: (82/53) 754-2666
Fax: (82/53) 752-4696

Samsung Hewlett-Packard Co. Ltd.
8/F, Daehankyoyuk Insurance Bldg.
382-1 Sun Hwa-Dong
Chung-Gu
TEEJON 301-050
Tel: (82/42) 256-6725
Fax: (82/42) 256-6727

KUWAIT
Al-Khaldiya Trading & Contracting
P.O. Box 830
SAFAT 13009
Tel: (965) 242 49 10, 241 17 26
Fax: (965) 245 20 67
Telex: 22481 AREEG KT
Cable: VISCOUNT

LEBANON
Computer Information Systems S.A.L.
Chammas Building
P.O. Box 11-6274
BEIRUT-BEIRUT
Tel: (961/1) 89 31 13
Fax: (961/1) 58 18 34
Telex: 42309 chacis le

LUXEMBOURG
Hewlett-Packard Belgium S.A./N.V.
Blvd de la Woluwe, 100
Woluwedal
1200 **BRUSSELS**
Tel: (32/2) 761-3111
Fax: (32/2) 763-0613
Telex: 23 494 palobe b

MALAWI
Field Consolidated (Private) Ltd.
System Division
Manhattan Court
61 Second Street
P.O. Box 3458
HARARE
Zimbabwe
Tel: (263/4) 73 98 81
Fax: (263/4) 70 20 08
Telex: 0907 26241

MALAYSIA
Hewlett-Packard Sales (M) SDN BHD
9/F, Chung Khiaw Bank Building
Section 46, Jalan Raja Laut
GPO Box 11119
50736 **KUALA LUMPUR**
Tel: (60/3) 298-6555
Fax: (60/3) 291-5495

Hewlett-Packard Sales (M) SDN BHD
6/F, Wisma Penang Garden
42, Jalan Sultan Ahmad Shah
10050 **PENANG**
Tel: (60/4) 376-532
Fax: (60/4) 371-027

MEXICO
Hewlett-Packard de Mexico,
S.A. de C.V.
Rio Nio No. 4049 Desp. 12
Fracc. Cordoba
CIUDAD JUAREZ, Mexico
Tel: (52/161) 3 15 62

Hewlett-Packard de Mexico,
S.A. de C.V.
Condominio Kadereyta
Circuito del Mezon No. 186 Desp. 6
Col. Del Prado
76030 **QUERETARO,** QRO
Tel: (52/4) 636-0271

Hewlett-Packard de Mexico,
S.A. de C.V.
Czda. del Valle
409 Ote. 4th Piso
Colonia del Valle
Municipio de Garcia
66220 **GARZA GARCIA,** N.L.
Tel: (52/8) 378-4240
Telex: 382410 HPMY

Hewlett-Packard de Mexico,
S.A. de C.V.
Monti Morelos No. 299
Fraccionamiento Loma Bonita 45060
GUADALAJARA, Jalisco
Tel: (52/3) 631-4600
Telex: 0684 186 ECOME

Hewlett-Packard de Mexico,
S.A. de C.V.
Monte Pelvoux No. 111
Lomas de Chapultepec
11000 **MEXICO,** D.F.
Tel: (52/5) 202-0155
Fax: (52/5) 540-4208

Hewlett-Packard de Mexico,
S.A. de C.V.
Blvd. Independencia
No. 2000 Ote. Piso 3
Col. Estrella
27010 **TORREON,** COA.
Tel: (52/1) 718-2201

MOROCCO
SICOTEL
Complexe des Habous
Tour C, avenue des Far
CASABLANCA 01
Tel: (212) 31 22 70
Telex: 0407 27604

NETHERLANDS
Hewlett-Packard Nederland B.V.
Startbaan 16
P.O. Box 667
1180 NL **AMSTELVEEN**
Tel: (31/20) 547 69 11
Fax: (31/20) 471 825
Telex: 13 216 HEPA NL

SALES OFFICES
Arranged alphabetically by country (cont'd)

NETHERLANDS (cont'd)

Hewlett-Packard Nederland B.V.
Boschdijk 131
P.O. Box 2342
5612 HB **EINDHOVEN**
Tel: (31/40) 32 69 11
Fax: (31/40) 44 65 46
Telex: 51484 hepae nl

NEW ZEALAND

Hewlett-Packard (N.Z.) Ltd.
5 Owens Road
Epsom, **AUCKLAND** 3
Tel: (64/9) 687-159
Fax: (64/9) 600-507

Hewlett-Packard (N.Z.) Ltd.
186-190 Willis Street
WELLINGTON 1
Tel: (64/4) 820-400
Fax: (64/4) 843-380

NIGER

Engineering Business Concept (E.B.C.)
21, Route du Canal
Zone 3
04 B.P. 1357
ABIDJAN
Ivory Coast
Tel: (225) 21 50 24
Fax: (225) 35 37 90

NIGERIA

Management Information Systems Co., Ltd.
3, Gerrard Road, Ikoyi
LAGOS
Tel: (234/1) 68 08 87
Fax: (234/1) 68 54 87
Telex: 23582

NORTHERN IRELAND
See United Kingdom

NORWAY

Hewlett-Packard Norge A/S
Oesterndalen 16-18
P.O. Box 34
N-1345 **OESTERAAS**
Tel: (47/2) 24-60 90
Fax: (47/2) 24 66 96, 24 30 78
Telex: 76621 HPNAS N

Hewlett-Packard Norge A/S
Boernergt. 42
Box 2470
N-5037 **SOLHEIMSVIK**
Tel: (47/5) 29 00 90
Fax: (47/5) 29 10 72

OMAN

Imtac LLC
P.O. Box 9196
MINA AL FAHAL
Tel: (968) 70-77-27, 70-77-23
Fax: (968) 79 66 39
Telex: 0498 3865

PAKISTAN

Mushko Electronics (PVT) Ltd.
68-W Sama Plaza
Blue Area, F-7
ISLAMABAD
Tel: (92/51) 814 492
Fax: (92/51) 821 201
Telex: 54001 Muski Pk

Mushko Electronics (PVT) Ltd.
Victoria Chambers
Abdullah Haroon Road
KARACHI
Tel: (92/21) 524 131, 524 132
Fax: (92/21) 512 298
Telex: 2894 MUSKO PK

PANAMA

Electronico Balboa, S.A.
Calle Samuel Lewis, Ed. Alfa
Apartado 4929
PANAMA CITY
Tel: (507) 636 613
Fax: (507) 361 820
Telex: 368 3483 ELECTRON PG

PERU

Cia Electro Médica, S.A.
Los Flamencos No. 145, Ofna. 301 & 302
San Isidro, **LIMA** 1
Tel: (51/14) 41-4325, 41-3703
Fax: (51/14) 42-3572
Telex: 39425257 PE PB SIS

PHILIPPINES

The Online Advanced Systems Corp.
2nd Floor, Electra House
115-117 Esteban Street
Legaspi Village, Makati
Metro **MANILA**
Tel: (63/2) 818-4394
Fax: (63/2) 816-0590
Telex: 63274 ONLINE PN

PORTUGAL

Hewlett-Packard Portugal S.A.
Torre de Santo Antonio
Rua Gregorio Lopes, Lote 1732B
Restelo
1400 **LISBON**
Tel: (351) 617 343/44/46
Fax: (351) 617345
Telex: 43126

Hewlett-Packard Portugal S.A.
Edificio Tranquilidade
Rua D. Manuel, 296
4000 **PORTO**
Tel: (351) 493 122
Telex: 26054
Fax: (351) 488 721

PUERTO RICO

Hewlett-Packard Puerto Rico
P.O. Box 4048
AGUADILLA 00605
Tel: (809) 890-6000

Hewlett-Packard Puerto Rico
101 Munoz Rivera Avenue
Esu. Calle Ochoa
HATO REY, 00918
Tel: (809) 754-7800

QATAR

Qatar Datamation Systems
P.O. Box 350
DOHA
Tel: (974) 44 19 16
Fax: (974) 44 12 19
Telex: 4833

SAUDI ARABIA

Modern Electronic Establishment
P.O. Box 281
Thouqbah
AL-KHOBAR 31952
Tel: (966/3) 895-1760, 895-1764
Telex: 0495 671 106 HPMEEK SJ
Cable: ELECTA AL-KHOBAR

Modern Electronic Establishment
P.O. Box 1228
Redec Plaza, 6th Floor
JEDDAH 21431
Tel: (966/2) 644 96 28
Fax: (966/2) 644 18 33
Cable: ELECTA JEDDAH

Modern Electronic Establishment
P.O.Box 22015
RIYADH 11495
Tel: (966/1) 476 30 30
Fax: (966/1) 476 76 49
Telex: 402049 MEERYD SJ

SCOTLAND
See United Kingdom

SINGAPORE

Hewlett-Packard Singapore Ltd.
150 Beach Road
#29-00 Gateway West
PO Box 2, Bras Basah Post Office
SINGAPORE, 9118
Tel: (65) 291 9088
Fax: (65) 292 7089
Telex: RS 34209

SOUTH AFRICA

HI Performance Systems (Pty.) Ltd.
P.O. Box 120, Howard Place
CAPE TOWN 7450
Tel: (27/21) 53-7954
Fax: (27/21) 53-5119

Hi Performance Systems (Pty.) Ltd.
Private Bag Wendywood
SANDTON 2144
Tel: (27/11) 802-5111
Fax: (27/11) 802-6332

SPAIN

Barcelona

Hewlett-Packard Española S.A.
Avda. Diagonal, 605
08028 **BARCELONA**
Tel: (34/3) 401 91 00
Fax: (34/3) 230 84 68
Telex: 52603 hpbee

Bilbao

Hewlett-Packard Española, S.A.
Avda. de Zugazarte, 8
(Edificio El Abra 4)
48930 **LAS ARENAS-GUECHO**
Tel: (34/4) 464 32 55
Fax: (34/4) 464 90 83
Telex: 33032

Madrid

Hewlett-Packard Española S.A.
Crta. de la Coruña, km 16,500
28230 Las Rozas
MADRID
Tel: (34/1) 637 00 11
Fax: (34/1) 637 55 62
Telex: 23515 HPE

Sevilla

Hewlett-Packard Española S.A.
c/Luis de Morales, S/N
Edificio Forum
41005 **SEVILLA**
Tel: (34/5) 458 17 00
Fax: (34/5) 457 57 20
Telex: 72933

Valencia

Hewlett-Packard Española S.A.
Isabel La Católica, 8
46004 **VALENCIA**
Tel: (34/6) 351 59 44
Telex: 63435
Fax: (34/6) 351 49 06

SWEDEN

Göteborg

Hewlett-Packard Sverige AB
Topasgatan 1A
Box 266
42123 **VÄSTRA-FRÖLUNDA**
Tel: (46/31) 89 10 00
Fax: (46/31) 49 50 99

Malmö

Hewlett-Packard Sverige AB
Östra Tullgatan 3
20011 **MALMÖ**
Box 6132
Tel: (46/40) 702 70
Fax: (46/40) 97 74 18

Örebro

Hewlett-Packard Sverige AB
Elementvägen 16
70227 **ÖREBRO**
Tel: (46/19) 10 48 80
Fax: (46/19) 27 01 87

Stockholm

Hewlett-Packard Sverige AB
Skalholtsgatan 9, Kista
P.O. Box 19
16493 **KISTA**
Tel: (46/8) 750 20 00
Fax: (46/8) 752 77 81
Telex: (854) 17885

SWITZERLAND

Basel
Hewlett-Packard (Schweiz) AG
Clarastrasse 12
4058 **BÂLE**
Tel: (41/61) 681 59 20
Fax: (41/61) 681 98 59
Telex: 964 719

Geneva/Meyrin
Hewlett-Packard (Schweiz) AG
39, rue de Veyrot
1217 **MEYRIN** 1
Tel: (41/22) 780 41 11
Fax: (41/22) 782 13 68
Telex:27333 HPAG CH

Widen
Hewlett-Packard (Schweiz) AG
Allmend 2
8967 **WIDEN**
Tel: (41/57) 32 11 11
Telex: 828 010 hpag ch
Fax: (41/57) 31 73 74

TAIWAN
Hewlett-Packard Taiwan Ltd.
11/F, 456, Chung Hsiao 1st Road
KAOHSIUNG
Tel: (886/7) 241-2318

Hewlett-Packard Taiwan Ltd.
THM Office
20, Kao-Shuang Road
PIN CHEN, Taoyuan
Tel: (886/3) 492-9660
Fax: (886/3) 492-8087

Hewlett-Packard Taiwan Ltd.
126-1, Sec. 2
Lu-Shung Road
TAICHUNG
Tel: (886/4) 237-1425

Hewlett-Packard Taiwan Ltd.
8/F, Hewlett-Packard Building
337 Fu Hsing North Road
TAIPEI
Tel: (886/2) 712-0404
Fax: (886/2) 715-0851

TANZANIA
Adcom Ltd, Inc. Kenya
P.O. Box 30070
NAIROBI
Kenya
Tel: (254/2) 33 19 55
Telex: 0987 22639

THAILAND
Hewlett-Packard Thailand Ltd.
10th floor
Pacific Place
140 Sukhumvit Road
BANGKOK 10110
Tel: (66/2) 254-6720
Fax: (66/2) 254-6731
Telex: 086 84439 Simonco TH
Cable: UNIMESA Bangkok

TOGO
Engineering Business Concept (E.B.C)
21, Route du Canal
Zone 3
04 B.P. 1357
ABIDJAN
Ivory Coast
Tel: (225) 21 50 24
Fax: (225) 25 27 90

TRINIDAD & TOBAGO
Computer and Controls Ltd.
80-82 Edwards Street
P.O. Box 51
PORT-OF-SPAIN, Trinidad
Tel: (809) 624-2388/2389
Fax: (809) 625-6547

TUNISIA
Precision Electronique
5, rue de Chypre
Mutuelleville
1002 **TUNIS BELVEDERE**
Tunisia
Tel: (216/1) 78 50 37
Telex: 0409 13238

TURKEY
Hewlett-Packard Bilgisayar Ve Olcum Sistemleri A.S.
Paris Caddesi No 3
Diare 9
06670 **ANKARA**
Tel: (90/4) 125 83 13
Fax: (90/4) 125 47 45
Telex: 0607 46180

Hewlett-Packard Bilgisayar Ve Olcum Sistemleri A.S.
Mesrutiyet Mah, 19 Mayis Cad.
Nova-Baran Plaza Kat: 11-12
ISLI / ISTANBUL
Tel: (901) 175 29 70
Fax: (901) 175 29 92
Telex: 0607 39150

UGANDA
Adcom Ltd, Inc. Kenya
P O Box 30070
NAIROBI
Kenya
Tel: (254/2) 33 19 55
Telex: 0987 22639

UNION OF SOVIET SOCIALIST REPUBLICS
Hewelt-Packard Representative Office
Pokrovski Blvd. 4/17 KV12
10100 **MOSCOW**
Tel: (007/095) 923 5001
Fax: (007/095) 230 2611

UNITED ARAB EMIRATES
Emitac Ltd.
P.O. Box 2711
ABU DHABI
Tel: (971/2) 770419
Fax: (971/2) 72 30 58

Emitac Ltd.
Block "B" Arenco Bldg.
Zabel Road
P.O. Box 8391
DUBAI
Tel: (971/4) 37 75 91
Fax: (971/4) 37 08 99

Emitac Ltd.
P.O. Box 1641
SHARJAH
Tel: (971/6) 591181
Telex: 0893 48710 EMITAC EM
Cable: EMITAC SHARJAH

UNITED KINGDOM

ENGLAND
Birmingham
Hewlett-Packard Limited
Avon House
435 Stratford Road
Shirley, **SOLIHULL**
West Midlands B90 4BL
Tel: (44/21) 745-8800
Fax: (44/21) 733-5104
Telex: 339 105

Bracknell
Hewlett-Packard Limited
Coln Road
BRACKNELL
Berkshire RG 12 1HN
Tel: (44/344) 36 0000
Fax: (44/344) 36 3344
Telex: 849 555

Central London
Hewlett-Packard Limited
Bridewell House
9 Bridewell Place
LONDON EC4V 6BS
Tel: (44/71) 583-6565
Fax: (44/71) 583-6565
Ext. 5713/57803
Telex: 298163

Gatwick
Hewlett-Packard Limited
Churchill Court
Gatewick Road
CRAWLEY
West Sussex RH10 2PN
Tel: (44/293) 562955
Fax: (44/293) 578574
Telex: 878862

Manchester
Hewlett-Packard Limited
Heathside Park Road
Cheadle Heath
STOCKPORT
Cheshire SK3 ORB
Tel: (44/61) 428-0828
Fax: (44/61) 495-5009
Telex: 668 068

Uxbridge
Hewlett-Packard Limited
Harman House
No. 1 George Street
UXBRIDGE
Middlesex UB8 1YH
Tel: (44/895) 72020
Fax: (44/895) 75260
Telex: 893 135

Winnersh
Hewlett-Packard Limited
King Street Lane
WINNERSH Wokingham
Berkshire RG 11 5AR
Tel: (44/734) 78 4774
Fax: (44/734) 77 7285
Telex: 847 178

NORTHERN IRELAND
Hewlett-Packard (Ireland) Ltd.
Carrickfergus Industrial Centre
75 Belfast Road, **CARRICKFERGUS**
Co. Antrim BT38 8PM
Tel: (44/960) 367333

SCOTLAND
Hewlett-Packard Ltd.
1/3 Springburn Place
College Milton North
EAST KILBRIDE, G74 5NU
Tel: (44/35) 524 92 61
Fax: (44/35) 523 59 29
Telex: 779615

Hewlett-Packard Ltd.
SOUTH QUEENSFERRY
West Lothian, EH30 9TG
Tel: (44/31) 331 11 88
Fax: (44/31) 331

UNITED STATES
Hewlett-Packard Co.
Customer Information Center
Tel: (800) 752-0900
Hours: 8:00 AM to 5:00 PM
Pacific Time

Alabama
Hewlett-Packard Co.
2100 Riverchase Center
Building 100 - Suite 118
BIRMINGHAM, AL 35244
Tel: (205) 988-0547
Fax: (205) 988-5308

Hewlett-Packard Co.
620 Discovery Dr.
HUNTSVILLE, AL 35806
Tel: (205) 830-2000
Fax: (205) 830-1427

Alaska
Hewlett-Packard Co
4000 Old Seward Highway
Suite 101
ANCHORAGE, AK 99503
Tel (907) 563-8855
Fax: (907) 561-7409

SALES OFFICES
Arranged alphabetically by country (cont'd)

UNITED STATES (cont'd)

Arizona

Hewlett-Packard Co.
8080 Pointe Parkway West
PHOENIX, AZ 85044
Tel: (602) 273-8000
Fax: (602) 273-8080

Hewlett-Packard Co.
3400 East Britannia Dr.
Bldg. C, Suite 124
TUCSON, AZ 85706
Tel: (602) 573-7400
Fax: (602) 573-7429

Arkansas

Hewlett-Packard Co.
10816 Executive Center Dr
Conway Bldg. Suite 116
LITTLE ROCK, AR 72211
Tel: (501) 225-7178
Fax: (501) 221-3614

California

Hewlett-Packard Co.
26701 W. Argoura Rd.
CALABASAS, CA 91302
Tel: (818) 880-3400
Fax: (818) 880-3437

Hewlett-Packard Co.
353 Lakeside Dr
FOSTER CITY, CA 94404
Tel: (415) 378-8400
Fax: (415) 378-8405

Hewlett-Packard Co.
1907 North Gateway Blvd.
FRESNO, CA 93727
Tel: (209) 252-9652
Fax: (209) 456-9302

Hewlett-Packard Co.
1421 S. Manhattan Av.
FULLERTON, CA 92631
Tel: (714) 999-6700
Fax: (714) 778-3033

Hewlett-Packard Co.
7408 Hollister Ave. #A
GOLETA, CA 93117
Tel: (805) 685-6100
Fax: (805) 685-6163

Hewlett-Packard Co.
9800 Muirlando Ave.
IRVINE, CA 92718
Tel: (714) 472-3000
Fax: (714) 581-3607 (Direct Dial only)

Hewlett-Packard Co.
2525 Grand Avenue
LONG BEACH, CA 90815
Tel: (213) 498-1111
Fax: (213) 494-1986

Hewlett-Packard Co.
5651 West Manchester Ave.
LOS ANGELES, CA 90045
Tel: (213) 337-8000
Fax: (213) 337-8338

Hewlett-Packard Co.
351 E. Evelyn Ave.
Bldg. 33
MOUNTAIN VIEW, CA 94039
Tel: (415) 694-2000
Fax: (415) 694-0600

Hewlett-Packard Co.
5161 Lankershim Blvd.
NORTH HOLLYWOOD, CA 91601
Tel: (818) 505-5600
Fax: (818) 505-5875

Hewlett-Packard Co.
5725 W. Las Positas Blvd.
PLEASANTON, CA 94566
Tel: (415) 460-0282
Fax: (415) 460-0713

Hewlett-Packard Co.
4244 So. Market Court, Suite A
SACRAMENTO, CA 95834
Tel: (916) 929-7222
Fax: (916) 927-7152

Hewlett-Packard Co.
9606 Aero Drive
SAN DIEGO, CA 92123
Tel: (619) 279-3200
Fax: (619) 268-8487

Hewlett-Packard Co.
50 Fremont St. Suite 200
SAN FRANCISCO, CA 94105
Tel: (415) 882-6800
Fax: (415) 882-6805

SANTA BARBARA (see **GOLETA**)

Hewlett-Packard Co.
3003 Scott Boulevard
SANTA CLARA, CA 95054
Tel: (408) 988-7000
Fax: (408) 988-7103

Hewlett-Packard Co.
5280 Valentine Rd. Suite 205
VENTURA, CA 93003
Tel: (805) 658-6898
Fax: (805) 650-0721

Colorado

Hewlett-Packard Co.
3005 Center Green Drive #205
Suite A
BOULDER, CO 80301
Tel: (303) 530-3940
Fax: (303) 938-3025

Hewlett-Packard Co.
24 Inverness Place, East
ENGLEWOOD, CO 80112
Tel: (303) 649-5000
Fax: (303) 649-5787

Connecticut

Hewlett-Packard Co.
3 Parkland Dr.
DARIEN, CT 06820
Tel: (203) 656-0040
Fax: (203) 656-5563

Hewlett-Packard Co.
115 Glastonbury Blvd
GLASTONBURY, CT 06033
Tel: (203) 633-8100
Fax: (203) 659-6087

District of Columbia

(see Rockville, MD)

Florida

Hewlett-Packard Co.
5900 N. Andrews, Suite 100
FORT LAUDERDALE, FL 33309
Tel: (305) 938-9800
Fax: (305) 938-2293

Hewlett-Packard Co.
6800 South Point Parkway
Suite 301
JACKSONVILLE, FL 32216
Tel: (904) 636-9955
Fax: (904) 636-9955

Hewlett-Packard Co.
255 East Drive, Suite B
MELBOURNE, FL 32901
Tel: (407) 729-0704
Fax: (407) 723-4557

Hewlett-Packard Co.
6177 Lake Ellenor Drive
ORLANDO, FL 32809
Tel: (407) 859-2900
Fax: (407) 826-9309

Hewlett-Packard Co.
4700 Bayou Blvd.
Building 5
PENSACOLA, FL 32503
Tel: (904) 476-8422
Fax: (904) 476-4116

Hewlett-Packard Co.
5550 Idlewild, #150
TAMPA, FL 33634
Tel: (813) 884-3282
Fax: (813) 889-4445

Hewlett-Packard Co.
2015 South Park Place
ATLANTA, GA 30339
Tel: (404) 955-1500
Fax: (404) 980-7669

Hewlett-Packard Co.
3607 Parkway Lane
Suite 300
NORCROSS, GA 30092
Tel: (404) 448-1894
Fax: (404) 246-5206

Hawaii

Hewlett-Packard Co.
Pacific Tower
1001 Bishop St.
Suite 2400
HONOLULU, HI 96813
Tel: (808) 526-1555
Fax: (808) 536-7873

Idaho

Hewlett-Packard Co.
11309 Chinden Blvd.
BOISE, ID 83714
Tel: (208) 323-2700
Fax: (208) 323-2528

Illinois

Hewlett-Packard Co.
2205 E. Empire St.
BLOOMINGTON, IL 61704
Tel: (309) 662-9411
Fax: (309) 662-0351

Hewlett-Packard Co.
525 W. Monroe St., Suite 1308
CHICAGO, IL 60606
Tel: (312) 930-0010
Fax: (312) 930-0986

Hewlett-Packard Co.
1200 East Diehl Road
NAPERVILLE, IL 60566
Tel: (708) 505-8800
Fax: (708) 505-7876

Hewlett-Packard Co.
5201 Tollview Drive
ROLLING MEADOWS, IL 60008
Tel: (708) 255-9800
Fax: (708) 259-5878

Indiana

Hewlett-Packard Co.
11911 N. Meridian St.
CARMEL, IN 46032
Tel: (317) 844-4100
Fax: (317) 843-1291

Hewlett-Packard Co.
111 E. Ludwig Road
Suite 108
FT. WAYNE, IN 46825
Tel: (219) 482-4283
Fax: (219) 482-9907

INDIANAPOLIS (see **CARMEL**)

Iowa

Hewlett-Packard Co.
4050 River Center Court
CEDAR RAPIDS, IA 52402
Tel: (319) 393-0606
Fax: (319) 378-1024

Hewlett-Packard Co.
4201 Corporate Dr.
WEST DES MOINES, IA 50265
Tel: (515) 224-1435
Fax: (515) 224-1870

Kansas

Hewlett-Packard Co.
North Rock Business Park
3450 N. Rock Rd.
Suite 300
WICHITA, KS 67226
Tel: (316) 636-4040
Fax: (316) 636-4504

Kentucky

Hewlett-Packard Co.
305 N. Hurstbourne Lane,
Suite 100
LOUISVILLE, KY 40222
Tel: (502) 426-0100
Fax: (502) 426-0322

Louisiana

Hewlett-Packard Co.
160 James Drive East
ST. ROSE, LA 70087
Tel: (504) 467-4100
Fax: (504) 467-4100 x 291

Maryland

Hewlett-Packard Co.
3701 Koppers Street
BALTIMORE, MD 21227
Tel: (301) 644-5800
Fax: (301) 362-7650

Hewlett-Packard Co.
2 Choke Cherry Road
ROCKVILLE, MD 20850
Tel: (301) 258-2000
Fax: (301) 258-5986

Massachusetts

Hewlett-Packard Co.
1775 Minuteman Road
ANDOVER, MA 01810
Tel: (508) 682-1500
Fax: (508) 794-2619

Hewlett-Packard Co.
29 Burlington Mall Rd.
BURLINGTON, MA 01803-4514
Tel: (617) 270-7000
Fax: (617) 221-5240

Michigan

Hewlett-Packard Co.
3033 Orchard Vista S.E.
GRAND RAPIDS, MI 49546
Tel: (616) 957-1970
Fax: (616) 956-9022

Hewlett-Packard Co.
39550 Orchard Hill Place Drive
NOVI, MI 48050
Tel: (313) 349-9200
Fax: (313) 349-9240

Hewlett-Packard Co.
560 Kirts Rd.
Suite 101
TROY, MI 48084
Tel: (313) 362-5180
Fax: (313) 362-3028

Minnesota

Hewlett-Packard Co.
2025 W. Larpenteur Ave.
ST. PAUL, MN 55113
Tel: (612) 644-1100
Fax: (612) 641-9787

Mississippi

Hewlett-Packard Co.
800 Woodland Parkway, Suite 101
RIDGELAND, MS 39157
Tel: (601) 957-0730
Fax: (601) 957-2515

Missouri

Hewlett-Packard Co.
6601 Winchester Ave.
KANSAS CITY, MO 64133
Tel: (816) 737-0071
Fax: (816) 737-4690

Hewlett-Packard Co.
530 Maryville Centre Drive
ST. LOUIS, MO 63140
Tel: (314) 542-1500
Fax: (314) 542-1586

Nebraska

Hewlett-Packard
11626 Nicholas St.
OMAHA, NE 68154
Tel: (402) 493-0300
Fax: (402) 493-4334

New Jersey

Hewlett-Packard Co.
120 W. Century Road
PARAMUS, NJ 07653
Tel: (201) 599-5000
Fax: (201) 599-5382

Hewlett-Packard Co.
10 Sylvan Way
PARSIPPANY, NJ 07054
Tel: (201) 682-4000
Fax: (201) 682-4031

Hewlett-Packard Co.
20 New England Ave. West
PISCATAWAY, NJ 08854
Tel: (201) 562-6100
Fax: (201) 562-6246

New Mexico

Hewlett-Packard Co.
7801 Jefferson N.E.
ALBUQUERQUE, NM 87109
Tel: (505) 823-6100
Fax: (505) 823-1243

Hewlett-Packard Co.
1362-C Trinity Dr.
LOS ALAMOS, NM 87544
Tel: (505) 662-6700
Fax: (505) 662-4312

New York

Hewlett-Packard Co.
5 Computer Drive South
ALBANY, NY 12205
Tel: (518) 458-1550
Fax: (518) 458-1550 x 0393

Hewlett-Packard Co.
130 John Muir Dr.
AMHERST, NY 14228
Tel: (716) 689-3003
Fax: (716) 636-7034

BUFFALO (see AMHERST)

Hewlett-Packard Co.
200 Cross Keys Office Park
FAIRPORT, NY 14450
Tel: (716) 223-9950
Fax: (716) 223-6331

Hewlett-Packard Co.
7641 Henry Clay Blvd.
LIVERPOOL, NY 13088
Tel: (315) 451-1820
Fax: (315) 451-1820 x 255

Hewlett-Packard Co.
No. 1 Pennsylvania Plaza
55th Floor
34th Street & 7th Avenue
MANHATTAN NY 10119
Tel: (212) 971-0800
Fax: (212) 330-6967

Hewlett-Packard Co.
7 Old Sod Farm Road
MELVILLE, NY 11714
Tel: (516) 753-0555
Fax: (516) 753-3469

Hewlett-Packard Co.
2975 Westchester Ave
PURCHASE, NY 10577
Tel: (914) 935-6300
Fax: (914) 935-6497

ROCHESTER (see FAIRPORT)

SYRACUSE (see LIVERPOOL)

Hewlett-Packard Co.
Executive Square Office Bldg.
66 Middlebush Rd.
WAPPINGERS FALLS, NY 12590
Tel: (914) 298-9125
Fax: (914) 298-9154

North Carolina

Hewlett-Packard Co.
305 Gregson Dr.
CARY, NC 27511
Tel: (919) 467-6600
Fax: (919) 460-2296
(919) 460-2297

Hewlett-Packard Co.
P.O. Box 240318
CHARLOTTE, NC 28224
Tel: (704) 527-8780
Fax: (704) 523-7857

Hewlett-Packard Co.
7029 Albert Pick Rd. #100
GREENSBORO, NC 27409
Tel: (919) 665-1800
Fax: (919) 668-1797
Mailing Address
PO Box 26500
Greensboro, NC 27426

RALEIGH (see CARY)

Ohio

AKRON (see COPLEY)

Hewlett-Packard Co.
4501 Erskine Road
CINCINNATI, OH 45242
Tel: (513) 891-9870
Fax: (513) 891-0033

CLEVELAND (see STRONGSVILLE)

Hewlett-Packard Co.
Montrose West Ave.
COPLEY, OH 44321
Tel: (216) 666-7711
Fax: (216) 666-6054

Hewlett-Packard Co.
7887 Washington Village Dr.
DAYTON, OH 45459
Tel: (513) 433-2223
Fax: (513) 433-8633

Hewlett-Packard Co.
15885 Sprague Road
STRONGSVILLE, OH 44136
Tel: (216) 243-7300
Fax: (216) 234-7230

Hewlett-Packard Co.
One Maritime Plaza, 5th Floor
720 Water Street
TOLEDO, OH 43604
Tel: (419) 242-2200
Fax: (419) 241-7655

Hewlett-Packard Co.
675 Brooksedge Blvd.
WESTERVILLE, OH 43081
Tel: (614) 891-3344
Fax: (614) 891-1476

Oklahoma

Hewlett-Packard Co.
3525 N.W. 56th St.
Suite C-100
OKLAHOMA CITY, OK 73112
Tel: (405) 946-9499
Fax: (405) 942-2127

Hewlett-Packard Co.
6655 South Lewis,
Suite 105
TULSA, OK 74136
Tel: (918) 481-6700
Fax: (918) 481-2250

Oregon

EUGENE (see WILSONVILLE)

Hewlett-Packard Co.
9255 S. W. Pioneer Court
WILSONVILLE, OR 97070
Tel: (503) 484-4085
Fax: (503) 682-8155

Pennsylvania

Hewlett-Packard Co.
Heatherwood Industrial Park
50 Dorchester Rd.
P.O. Box 6080
HARRISBURG, PA 17112
Tel: (717) 657-5900
Fax: (717) 657-5946

Hewlett-Packard Co.
111 Zeta Drive
PITTSBURGH, PA 15238
Tel: (412) 782-0400
Fax: (412) 963-1300

Hewlett-Packard Co.
2750 Monroe Boulevard
VALLEY FORGE, PA 19482
Tel: (215) 666-9000
Fax: (215) 666-2034

South Carolina

Hewlett-Packard Co.
Brookside Park, Suite 122
1 Harbison Way
COLUMBIA, SC 29212
Tel: (803) 732-0400
Fax: (803) 732-4567

Hewlett-Packard Co.
545 N. Pleasantburg Dr.
Suite 100
GREENVILLE, SC 29607
Tel: (803) 232-8002
Fax: (803) 232-8739

Tennessee

Hewlett-Packard Co.
One Energy Center Suite 200
Pellissippi Pkwy.
KNOXVILLE, TN 37932
Tel: (615) 966-4747
Fax: (615) 966-8147

SALES OFFICES
Arranged alphabetically by country (cont'd)

UNITED STATES (cont'd)

Tennessee (cont'd)
Hewlett-Packard Co.
889 Ridge Lake Blvd.,
Suite 100
MEMPHIS, TN 38119
Tel: (901) 763-4747
Fax: (901) 762-9723

Hewlett-Packard Co.
44 Vantage Way,
Suite 160
NASHVILLE, TN 37228
Tel: (615) 255-1271
Fax: (615) 726-2310

Texas
Hewlett-Packard Co.
9050 Capital of Texas Highway, North
#290
AUSTIN, TX 78759
Tel: (512) 346-3855
Fax: (512) 338-7201
Mailing Address
PO Box 9431
Austin, TX 78766-9430

DALLAS (see **RICHARDSON**)

Hewlett-Packard Co.
5700 Cromo Dr
EL PASO, TX 79912
Tel: (915) 833-4400
Fax: (915) 581-8097

Hewlett-Packard Co.
10535 Harwin Drive
HOUSTON, TX 77036
Tel: (713) 776-6400
Fax: (713) 776-6495

Hewlett-Packard Co.
3301 West Royal Lane
IRVING, TX 75063
Tel: (214) 869-3377
Fax: (214) 830-8951

Hewlett-Packard Co.
109 E. Toronto, Suite 100
McALLEN, TX 78503
Tel: (512) 630-3030
Fax: (512) 630-1355

Hewlett-Packard Co.
930 E. Campbell Rd.
RICHARDSON, TX 75081
Tel: (214) 231-6101
Fax: (214) 699-4337

Hewlett-Packard Co.
14100 San Pedro Ave., Suite 100
SAN ANTONIO, TX 78232
Tel: (512) 494-9336
Fax: (512) 491-1299

Utah
Hewlett-Packard Co.
3530 W. 2100 South
SALT LAKE CITY, UT 84119
Tel: (801) 974-1700
Fax: (801) 974-1780

Virginia
Hewlett-Packard Co.
840 Greenbrier Circle
Suite 101
CHESAPEAKE, VA 23320
Tel: (804) 424-7105
Fax: (804) 424-1494

Hewlett-Packard Co.
4401 Water Front Dr.
GLEN ALLEN, VA 23060
Tel: (804) 747-7750
Fax: (804) 965-9297

NORFOLK (see **CHESAPEAKE**)

RICHMOND (see **GLEN ALLEN**)

Hewlett-Packard Co.
2800 Electric Road Suite 100
ROANOKE, VA 24018
Tel: (703) 774-3444
Fax: (703) 989-8049

Washington
Hewlett-Packard Co.
15815 S.E. 37th Street
BELLEVUE, WA 98006
Tel: (206) 643-4000
Fax: (206) 643-8748

Hewlett-Packard Co.
N. 1225 Argonne Rd
SPOKANE, WA 99212-2657
Tel: (509) 922-7000
Fax: (509) 927-4236

Washington, D.C.
(see Rockville, MD)

West Virginia
Hewlett-Packard Co.
501 56th Street
CHARLESTON, WV 25304
Tel: (304) 925-0492
Fax: (304) 925-1910

Wisconsin
Hewlett-Packard Co.
275 N. Corporate Dr.
BROOKFIELD, WI 53005
Tel: (414) 792-8800
Fax: (414) 792-0218

Hewlett-Packard Co.
333 Main St.
GREEN BAY, WI 54301
Tel: (414) 436-2780
Fax: (414) 436-2786

MILWAUKEE (see **BROOKFIELD**)

URUGUAY
Pablo Ferrando S.A.C. e I.
Avenida Italia 2877
Casilla de Correo 370
MONTEVIDEO
Tel: (598/2) 802 586
Telex: 398802586

VENEZUELA
Hewlett-Packard de Venezuela, C.A.
Tercera Transversal
Los Ruices Norte
Edificio Segre, **CARACAS**
Tel: (58/2) 239 4133/4244
Fax: (58/2) 239 3080

YUGOSLAVIA
Do Hermes
General Zdanova 4
YU-11000 **BEOGRAD**
Tel: (38/11) 342 641
Telex: 11433

Do Hermes
Celovska 73
YU-61000 **LJUBLJANA**
Tel: (38/61) 553 170
Telex: 31583

ZAMBIA
R.J. Tilbury (Zambia) Ltd.
P.O. Box 32792
LUSAKA
Tel: (260/1) 21 55 80
Telex: 0902 40128

ZIMBABWE
Field Consolidated (Private) Ltd.
Systron Division
Manhattan Court
61 Second Street
P.O. Box 3458
HARARE
Tel: (263/4) 73 98 81
Fax: (263/4) 70 20 08
Telex: 26241

Please send directory corrections to:
Test & Measurement Catalog, 54AF
Hewlett-Packard Company
5301 Stevens Creek Blvd.
P.O. Box 58059
Santa Clara, CA 95052-8059
Tel: (408) 553-7271
Fax: (408) 983-1006

September 1990

5959-6047

Printed in U.S.A.

OPERATING NOTICE

64700-92004

THIS PAGE MAY BE DISCARDED AFTER YOU PLACE THE DOCUMENT UNDER IT INTO THE BINDER.

Important Notice

For HP 64700 Series Documents

This notice briefly describes the modular HP 64700 Series product structure and how existing manuals are affected by the changes in the structure. Although each HP 64700 Series product is now "modular" (may be ordered separately), only some HP 64700 Series manuals that accompany those products have been updated to describe the modular product structure.

The purpose of this notice is to show you how to use existing manuals with the modular HP 64700 Series products.

Topics Described

In this notice, these topics are described:

- Modular Product Structure
- Organization of Manuals
- Documents Not Used in the Modular Structure
- Installation/Service Guide
- Concepts Manual
- Ordering Terminal Interface Manuals Separately
- Using Earlier Versions of Cards
- To Install and Use Your System

Modular Product Structure

You can now order individual HP 64700 Series products rather than an entire emulation and analysis system, if you want. Ordering components separately allows you to "mix and match" them to make a system that fits your development needs. The individual components that you can order include:

- HP 64700A card cage
- Emulator card sets for various processors
- Analyzers:
  - 64703A (64-channel emulation bus analyzer plus 16 channels of state/timing analysis)
  - 64704A (80-channel emulation bus analyzer)
  - 64706A (48-channel emulation bus analyzer)
- Memory cards for some processors
- HP 64701A LAN (Local Area Network) card

How it Affects Existing Manuals

Because the product structure is "modular", and some manuals remain unchanged, those manuals describe several topics that are no longer appropriate in the modular structure. These include:

Outdated Information in Older Documents	Corresponding New Information
Diagrams for original HP 64700 chassis.	HP 64700A card cage diagrams; refer to the *HP 64700 Installation/Service Guide.*
References to documents which are not shipped with modular HP 64700 series products.	Described later in this notice.
References to original product alpha characters.	Alpha characters for modular products are described in this notice.

Although some references to documents are now inaccurate, product functionality remains the same.

Updating Your System

If you are "mixing and matching" modular and original cards for emulators and analyzers in the HP 64700A card cage, be sure to update your software through the Flash EPROM Update as described in the *Installation/Service Guide.* Contact your Hewlett-Packard Representative for more information.

What We Mean When We Say "Modular"

Earlier versions of HP 64700 Series products were referred to as emulation/analysis systems because they were built as complete development systems. Today we provide the flexibility that allows you to "mix and match" HP 64700 Series components, because the components are now separate products. The separate components are referred to as "modular products."

For example, in the original HP 64700 product structure, 647nnAL represented a self-contained emulation/analysis system with external state/timing analysis probes. Now the components that make up 647nnAL can each be ordered separately. The following table shows an example:

System Product Ordered in Original Structure	Equivalent Components Ordered in the Modular Structure
HP 64749A	HP 64700A Card Cage HP 64749G 68332 Emulator Card HP 64704A Emulation Bus Analyzer Card
HP 64749AL	HP 64700A Card Cage HP 64749G 68332 Emulator Card HP 64703A Emulation Bus Analyzer Card with External State/Timing Analyzer

If you ordered an HP 64701A LAN card with your HP 64700 Series system product, it may have been installed in the HP 64700A card cage at the factory.

In the Original Structure

The type of emulation/analysis system was referred to by an alpha character at the end of the model number (HP 647nnA/B/C/D). In the original documentation, references to the emulator type still include these alpha characters (A, B, C, D). Remember, because you have received this notice, you have also received manuals that still use these original alpha characters.

In addition, the presence of an external state/timing analyzer was represented by an "L" at the end of the product number (HP 647nnAL). In the original documentation, references to the external analyzer are still denoted by the "L" suffix.

In the Modular Structure

Each component has a unique model number and alpha character. The type of emulator card set is referred to by a new alpha character at the end of the existing model number (HP 647nnF/G/H/J/K). In original documentation that is shipped with modular products, where you see references to the outdated alpha characters, please substitute the new alpha characters.

The external analyzer is now referred to as HP 64703A, as shown earlier in the table of equivalent components. This replaces references to the external state/timing analyzer as an "L" at the end of the product number in the original structure.

Ordering Original System Products

You can still order original emulation/analysis system products with the original structure alpha characters (A, AL, B, BL, C, CL, D, DL, for example), at least through February 1991.

Organization of Manuals

This is the documentation you will receive with HP 64700 Series modular products:

Product	Model Number	Documentation	Binder Width
Card cage	64700A	*Installation/Service Guide* *Support Services Guide*	1 ½ inch
Emulator card sets	647nnF/G/H/J/K	*Important Notice* * *Terminal Interface User's Guide* **	1 inch
		Concepts of Emulation and Analysis	
Analyzers	64703A 64704A 64706A	*Notice*	
YLO memory cards	64725A 64726A 64727A	*Notice*	
LAN card	64701A	*Notice*	
Softkey interfaces	647nnS Opt 004	*Important Notice* * *Emulator Softkey Interface User's Guide* ** *Analyzer Softkey Interface User's Guide* **	1 ½ inch
		Softkey Interface Reference **	1 ½ inch
PC interfaces	647nnS Opt 006	*Important Notice* * *Emulator PC Interface User's Guide* ** *Analyzer PC Interface User's Guide* **	1 ½ inch
		PC Interface Reference **	1 inch

* The notice will be included only with manuals that have not been rewritten to add usability improvements.

** These are existing manuals that haven't been rewritten for the modular structure.

NOTE: A Support Services guide accompanies each binder.

Documents Not Used in the Modular Structure

You received this notice because some or all of the documentation for your product has not been updated for the new modular product structure. Some of those documents contain references to original documents that are not shipped with the modular structured products. This table lists those outdated documents and where you can find comparable information in other documents now shipped with modular HP 64700 Series products.

Older Documents Not Used in the Modular Structure	New Location of Comparable Information
System Overview Manual	*Installation/Service Guide* *Emulator/Analyzer User's Guides* *Reference Manuals* *Concepts of Emulation and Analysis*
Hardware Installation and Configuration Manual	*Installation/Service Guide*
Softkey Interface Installation Notice	*Installation/Service Guide*
MS-DOS Software Applications Installation Manual (for PC Interface installation)	*Installation/Service Guide*
HP 64700 Manual Map	Information about the modular structure of manuals is contained in this notice and in the *Installation/Service Guide*. The modular product structure, along with new and better organized manuals, keeps the documentation structure simple enough so that a manual map should not be needed.
Glossary	*Concepts of Emulation and Analysis* *User's Guides* *Reference Manuals*
ROM Replacement Notice	Information on updating ROMs is included with the Flash EPROM Update. Refer to the *Installation/Service Guide* for more information.

Installation/Service Guide

The *Installation/Service Guide* covers everything you might need to know about installing, updating, and servicing HP 64700 series hardware, firmware, and software products. Some topics described in this manual include:

- HP 64700 Features
- Hardware Removal/Installation
- Software Installation
- System Configuration
- Updating Your System
- Service Information

In this guide, hardware removal and installation information covers emulator, analyzer, memory, LAN, and Flash EPROM cards, and the external state/timing analyzer probes. The section on software installation describes the Softkey Interface, along with details about connecting to the host computer and setting up LAN. That section also describes installing the PC Interface.

Also, because the design of the HP 64700A card cage differs from the original HP 64700 chassis, detailed diagrams of the HP 64700A are provided.

Concepts Manual

For those of you who want to "get an inside look" into emulation and analysis, a *Concepts of Emulation and Analysis* manual is now shipped with HP 64700 series modular emulator card sets (and with assembled emulation and analysis "system products" that contain an emulator card set). This manual explains how emulators work and how they can be used in advanced design tasks.

Ordering Terminal Interface Manuals Separately

When you order an emulator, you will also receive an *Emulator Terminal Interface User's Guide* written specifically for the emulator card set you ordered. This is the only Terminal Interface manual that you will receive automatically with your emulator.

You may order other Terminal Interface manuals as a separate package. That package includes the *Analyzer Terminal Interface User's Guide*, *Coordinated Measurement Bus (CMB) User's Guide*, and the *Terminal Interface Reference*. Those manuals describe additional features of the built-in HP 64700 Terminal Interface that are not documented in the *Emulator Terminal Interface User's Guide*.

By not shipping these manuals automatically, Hewlett-Packard reduces your cost and the number of excess documents that you may not want or need. This allows you to "tailor" your order to fit your specific needs.

If you ordered the optional Terminal Interface documentation package, you received:

Terminal Interface Documentation Package	Binder Width
Analyzer Terminal Interface User's Guide ** *CMB Terminal Interface User's Guide* **	1 1/2 inch
Terminal Interface Reference **	1 1/2 inch

** These existing manuals have not been rewritten for the modular structure.

NOTE: A Support Services guide accompanies each binder.

How to Order

If you did not order the Terminal Interface documentation package, but want those manuals, you must order the part number for that package. For more information, see your Hewlett-Packard Sales Representative.

Using Earlier Versions of Cards

You can use earlier versions of HP 647nn(A,B,C,D) emulator and analyzer cards in the HP 64700A card cage. **If you choose to do this, or if you "mix and match" original and modular cards in the HP 64700A card cage, you must update your system using the Flash EPROM Update.**

For more information, refer to the *HP 64700 Installation/Service Guide* or contact your Hewlett-Packard Representative. Or, you may call the Hewlett-Packard Microprocessor Development hotline free of charge at 1-800-447-3282 (in the United States).

To Install and Use Your System

Refer to the *HP 64700 Installation/Service Guide* for information about installing system components (including original and modular cards) and about connecting your HP 64700 Series system to a host computer or PC.

Refer to your *Softkey Interface, PC Interface,* or *Terminal Interface User's Guide* and *Reference* for details about using your HP 64700 Series emulator and analyzer.

Connecting Emulator Cards Together

When you order an emulator, you will receive one or more cards. Make sure that you connect the cards together correctly when installing them in the HP 64700A card cage, by observing the markings on the cables and connectors. For more information, refer to the *HP 64700 Installation/Service Guide.*

Using LAN

When using your emulator on the LAN, you must modify the file /usr/hp64000/etc/64700tab.net For more information about using LAN, your *HP 64700 Installation/Service Guide.*

Solving Problems

Refer to the *HP 64700 Installation/Service Guide* for information about solving installation problems with your emulator, analyzer, LAN, or the HP 64700A Card Cage.

Notes